Titre: Title:	Weakly-Labeled Data and Identity-Normalization for Facial Image Analysis
Auteur: Author:	David Rim
Date:	2013
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Rim, D. (2013). Weakly-Labeled Data and Identity-Normalization for Facial Image Analysis [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1335/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1335/
Directeurs de recherche: Advisors:	Christopher J. Pal
Programme: Program:	Génie informatique

UNIVERSITÉ DE MONTRÉAL

WEAKLY-LABELED DATA AND IDENTITY-NORMALIZATION FOR FACIAL IMAGE ANALYSIS

DAVID RIM
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
SEPTEMBRE 2013

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

WEAKLY-LABELED DATA AND IDENTITY-NORMALIZATION FOR FACIAL IMAGE ANALYSIS

présentée par : RIM David
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. DESMARAIS Michel C., Ph.D, président
M. PAL Christopher J., Ph.D, membre et directeur de recherche
M. BENGIO Yoshua, Ph.D., membre
M. LEARNED-MILLER Erik G., Ph.D., membre

For my parents.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Chris Pal, who has provided constant guidance and encouragement throughout my studies both at the Ecole Polytechnique de Montreal and also at the University of Rochester. I honestly do not know where I would be without his patience and efforts.

I would also like to thank Yoshua Bengio, Erik Learned-Miller, Gideon Mann and Michel Desmarais for their guidance and time as well.

I thank Ubisoft for both financial support and for providing the helmet camera video data used for our high quality animation control experiments. I also thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for financial support.

Finally, I would like to express my gratitude to my family who have supported me more than I deserve and more than I can ever repay.

ABSTRACT

This thesis deals with improving facial recognition and facial expression analysis using weak sources of information. Labeled data is often scarce, but unlabeled data often contains information which is helpful to learning a model. This thesis describes two examples of using this insight.

The first is a novel method for face-recognition based on leveraging weak or noisily labeled data. Unlabeled data can be acquired in a way which provides additional features. These features, while not being available for the labeled data, may still be useful with some foresight. This thesis discusses combining a labeled facial recognition dataset with face images extracted from videos on YouTube and face images returned from using a search engine. The web search engine and the video search engine can be viewed as very weak alternative classifier which provide "weak labels."

Using the results from these two different types of search queries as forms of weak labels, a robust method for classification can be developed. This method is based on graphical models, but also encorporates a probabilistic margin. More specifically, using a model inspired by the variational relevance vector machine (RVM), a probabilistic alternative to transductive support vector machines (TSVM) is further developed. In contrast to previous formulations of RVMs, the choice of an Exponential hyperprior is introduced to produce an approximation to the L_1 penalty. Experimental results where noisy labels are simulated and separate experiments where noisy labels from image and video search results using names as queries both indicate that weak label information can be successfully leveraged.

Since the model depends heavily on sparse kernel regression methods, these methods are reviewed and discussed in detail. Several different sparse priors algorithms are described in detail. Experiments are shown which illustrate the behavior of each of these sparse priors. Used in conjunction with logistic regression, each sparsity inducing prior is shown to have varying effects in terms of sparsity and model fit. Extending this to other machine learning methods is straight forward since it is grounded firmly in Bayesian probability. An experiment in structured prediction using Conditional Random Fields on a medical image task is shown to illustrate how sparse priors can easily be incorporated in other tasks, and can yield improved results.

Labeled data may also contain weak sources of information that may not necessarily be used to maximum effect. For example, facial image datasets for the tasks of performance driven facial animation, emotion recognition, and facial key-point or landmark prediction often contain alternative labels from the task at hand. In emotion recognition data, for

example, emotion labels are often scarce. This may be because these images are extracted from a video, in which only a small segment depicts the emotion label. As a result, many images of the subject in the same setting using the same camera are unused.

However, this data can be used to improve the ability of learning techniques to generalize to new and unseen individuals by explicitly modeling previously seen variations related to identity and expression. Once identity and expression variation are separated, simpler supervised approaches can work quite well to generalize to unseen subjects. More specifically, in this thesis, probabilistic modeling of these sources of variation is used to "identity-normalize" various facial image representations. A variety of experiments are described in which performance on emotion recognition, markerless performance-driven facial animation and facial key-point tracking is consistently improved. This includes an algorithm which shows how this kind of normalization can be used for facial key-point localization.

In many cases in facial images, sources of information may be available that can be used to improve tasks. This includes weak labels which are provided during data gathering, such as the search query used to acquire data, as well as identity information in the case of many experimental image databases. This thesis argues in main that this information should be used and describes methods for doing so using the tools of probability.

RÉSUMÉ

Cette thèse traite de l'amélioration de la reconnaissance faciale et de l'analyse de l'expression du visage en utilisant des sources d'informations faibles. Les données étiquetées sont souvent rares, mais les données non étiquetées contiennent souvent des informations utiles pour l'apprentissage d'un modèle. Cette thèse décrit deux exemples d'utilisation de cette idée.

Le premier est une nouvelle méthode pour la reconnaissance faciale basée sur l'exploitation de données étiquetées faiblement ou bruyamment. Les données non étiquetées peuvent être acquises d'une manière qui offre des caractéristiques supplémentaires. Ces caractéristiques, tout en n'étant pas disponibles pour les données étiquetées, peuvent encore être utiles avec un peu de prévoyance. Cette thèse traite de la combinaison d'un ensemble de données étiquetées pour la reconnaissance faciale avec des images des visages extraits de vidéos sur YouTube et des images des visages obtenues à partir d'un moteur de recherche. Le moteur de recherche web et le moteur de recherche vidéo peuvent être considérés comme de classificateurs très faibles alternatifs qui fournissent des étiquettes faibles.

En utilisant les résultats de ces deux types de requêtes de recherche comme des formes d'étiquettes faibles différents, une méthode robuste pour la classification peut être développée. Cette méthode est basée sur des modèles graphiques, mais aussi incorporant une marge probabiliste. Plus précisément, en utilisant un modèle inspiré par la variational relevance vector machine (RVM), une alternative probabiliste à la support vector machine (SVM) est développée.

Contrairement aux formulations précédentes de la RVM, le choix d'une probabilité a priori exponentielle est introduit pour produire une approximation de la pénalité L_1. Les résultats expérimentaux où les étiquettes bruyantes sont simulées, et les deux expériences distinctes où les étiquettes bruyantes de l'image et les résultats de recherche vidéo en utilisant des noms comme les requêtes indiquent que l'information faible dans les étiquettes peut être exploitée avec succès.

Puisque le modèle dépend fortement des méthodes noyau de régression clairsemées, ces méthodes sont examinées et discutées en détail. Plusieurs algorithmes différents utilisant les distributions a priori pour encourager les modelés clairsemés sont décrits en détail. Des expériences sont montrées qui illustrent le comportement de chacune de ces distributions. Utilisés en conjonction avec la régression logistique, les effets de chaque distribution sur l'ajustement du modèle et la complexité du modèle sont montrés.

Les extensions aux autres méthodes d'apprentissage machine sont directes, car l'approche

est ancrée dans la probabilité bayésienne. Une expérience dans la prédiction structurée utilisant un conditional random field pour une tâche d'imagerie médicale est montrée pour illustrer comment ces distributions a priori peuvent être incorporées facilement à d'autres tâches et peuvent donner de meilleurs résultats.

Les données étiquetées peuvent également contenir des sources faibles d'informations qui ne peuvent pas nécessairement être utilisées pour un effet maximum. Par exemple les ensembles de données d'images des visages pour les tâches tels que, l'animation faciale contrôlée par les performances des comédiens, la reconnaissance des émotions, et la prédiction des points clés ou les repères du visage contiennent souvent des étiquettes alternatives par rapport à la tache d'internet principale. Dans les données de reconnaissance des émotions, par exemple, des étiquettes de l'émotion sont souvent rares. C'est peut-être parce que ces images sont extraites d'une vidéo, dans laquelle seul un petit segment représente l'étiquette de l'émotion. En conséquence, de nombreuses images de l'objet sont dans le même contexte en utilisant le même appareil photo ne sont pas utilisés. Toutefois, ces données peuvent être utilisées pour améliorer la capacité des techniques d'apprentissage de généraliser pour des personnes nouvelles et pas encore vues en modélisant explicitement les variations vues précédemment liées à l'identité et à l'expression. Une fois l'identité et de la variation de l'expression sont séparées, les approches supervisées simples peuvent mieux généraliser aux identités de nouveau. Plus précisément, dans cette thèse, la modélisation probabiliste de ces sources de variation est utilisée pour identité normaliser et des diverses représentations d'images faciales. Une variété d'expériences sont décrites dans laquelle la performance est constamment améliorée, incluant la reconnaissance des émotions, les animations faciales contrôlées par des visages des comédiens sans marqueurs et le suivi des points clés sur des visages.

Dans de nombreux cas dans des images faciales, des sources d'information supplémentaire peuvent être disponibles qui peuvent être utilisées pour améliorer les tâches d'intérêt. Cela comprend des étiquettes faibles qui sont prévues pendant la collecte des données, telles que la requête de recherche utilisée pour acquérir des données, ainsi que des informations d'identité dans le cas de plusieurs bases de données d'images expérimentales. Cette thèse soutient en principal que cette information doit être utilisée et décrit les méthodes pour le faire en utilisant les outils de la probabilité.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ACRONYMS AND ABBREVIATIONS

AU	Action Unit
CHM	Conditional Harmonic Mixing
EBGM	Elastic Bunch Graph Matching
EM	Expectation-Maximization
FACS	Facial Action Coding System
FPLBP	Four-Pass LBP
GEC	Generalized Expectation Criteria
GEM	Generalized EM Criteria
GJD	Gabor Jet Descriptor
HMM	Hidden Markov Model
ICA	Independent Components Analysis
ITML	Information-Theoretic Metric Learning
KLR	Kernel Logistic Regression
LBP	Local Binary Patterns
LDML	Logistic Discriminant-based Metric Learning
LFW	Labeled Faces in the Wild
MAP	Maximum A Priori
ML	Maximum Likelihood
ML-II	Maximum Likelihood, Type 2
NMF	Non-Negative Matrix Factorization
OSS	One-shot Similarity
PCA	Principal Components Analysis
PPCA	Probabilistic Principal Components Analysis
RVM	Relevance Vector Machine
SIFT	Scale-Invariant Feature Transform
SSL	Semi-Supervised Learning
SVM	Support Vector Machines
TPLBP	Three-Pass LBP
TSVM	Transductive Support Vector Machine

CHAPTER 1

INTRODUCTION

This thesis is about using machine learning for facial image tasks. Facial images are relatively easy to obtain, either as static images or sampled from video data. A wealth of useful information is stored in a facial image, for example, the identity and emotional state of the person depicted. For a number of reasons, it would be ideal to obtain such information automatically.

Artificial intelligence offers the possibility to do this using computation. One method of accomplishing this goal relies on obtaining a large data set of facial images and labeling them with the appropriate identity or emotional state. Then, the goal becomes to learn a function which maps the input image with the correct label.

However, obtaining labels for face images is time-consuming and expensive. However, for most computer vision tasks dealing with faces, very large amounts of labeled data are necessary to learn good functions.

Machine learning, and in particular, semi-supervised learning offers the possibility of avoiding much of the time-consuming and error-prone effort of labeling. However, in many instances, data are accompanied by additional information which may help for learning good functions.

In some cases, the additional information may be labels that are often incorrect. For example, these labels may come from an alternative classifier. In this case, the problem becomes how to incorporate these "noisy" or "weak" labels well.

In other cases, the data may have additional labels which are not a target. For example, for facial expression recognition, the data may come with identity labels. There is often additional or "side-information" which is not directly associated with the label, which may be useful for learning.

This thesis investigates the use of weak-supervision and leveraging side-information for facial image tasks, in particular face recognition and facial expression recognition.

This chapter introduces and defines these concepts, as well as outlines the objectives and main scientific hypotheses of the research. The final section presents the organization of the remainder of this proposal.

1.1 Definitions and concepts

1.1.1 Face Recognition

Face recognition is defined in this thesis as the action of determining the identity associated with an image of a human face. In the context of computer vision, this is defined as the task of classifying a test image by identity. A related problem is face verification, in which the goal is to test whether two images belong to the same person. In essence, this can be viewed as in terms of binary image pair classification – same or not same. In this thesis, the face recognition problem is the identity classification task.

1.1.2 Facial Expression Recognition

Facial expression recognition is defined as the action of determining the label associated with facial deformations usually associated with an emotive state, *i.e.*"anger," "disgust," "fear," "happiness," "sadness," "surprise," "contempt," *etc.*This should not be confused with facial action unit detection, which is defined here as the identification of the non-rigid deformation of the face associated with facial muscle groupings. Expression labels, in this thesis, is synonymous with the emotion being expressed.

1.1.3 Facial Key-point Localization

A key step in many face recognition and facial expression recognition tasks is the need to detect and localize certain parts of the face – for example, eyes, nose and mouth. Face key-point localization is defined here as the action of determining the location of specific points in the face, such as the corners of the eyes, mouth and other face parts. Key-point locations are often used to build features for expression and facial action unit recognition. They can also be an important preprocessing step, as the key-points can be used to align facial images to remove certain kinds of variation. Variation which is the result of head or camera movement is also known as pose variation, as it depends on the pose of the person relative to the position of the camera.

1.1.4 Semi-Supervised Learning

The previous three subsections introduce the problem domains addressed in this thesis. Machine learning is the branch of artificial intelligence which attempts to produce systems built by learning from examples. In contrast to creating a complex system of hand-coded rules, machine learning attempts to create these systems automatically using examples.

If the learning algorithm is intended to create an algorithm which maps data to labels, and the algorithm is augmented with labels, this type of machine learning is known as supervised learning. This is because the algorithm is given a form of supervision in the form of labels. In contrast, without access to available labels, learning is called unsupervised learning.

Semi-Supervised Learning (SSL) is a type of machine learning which incorporates both unlabeled and labeled data. Typically, SSL is used in situations where labels are difficult to obtain but unlabeled data is plentiful, as in most computer vision tasks. For example, in object recognition tasks, images which may or may not contain the object are easy to obtain, but images which certainly contain the object are more difficult to obtain or to label in sufficient quantity.

There exist a range of algorithms which lie between the traditional labeled and unlabeled setting which could also be described as semi-supervised. This thesis will present a "weakly labeled" model, in which the unlabeled data is treated as labeled by a noisy process. Other side-information can also be used in lieu or in addition to labels. This thesis will also investigate the latter case, in which the information is used to obtain better input representations.

1.1.5 Weakly-Labeled Learning

Weakly labeled learning is defined in this thesis as a type of SSL in which the unlabeled data is accompanied by unreliable labels. The weak labels are not necessarily correct. In this case, supervised learning procedures can still be applied but are not necessarily appropriate. Many semi-supervised methods do not necessarily account for the presence of this kind of additional information.

1.2 Contributions of the Research

There are three main research questions posed in this thesis. Primarily they are concerned with how labeled and unlabeled data can be used to improve automatic face recognition and facial expression recognition.

1.2.1 Research Questions

- How can sparse kernel classifiers be designed in a way that allows them to be easily extended for other tasks ?
- How can weakly-labeled data improve face recognition ?
- How can identity labels be used to improve facial expression recognition tasks ?

1.2.2 General Objective

The objective of the research proposed in this document is to investigate improving face recognition using weakly labeled data and improving facial expression recognition using identity information. The main argument is that data often contain information that is not a feature or a label, which can and should be used effectively.

In facial recognition, this thesis seeks to show that incorporating weak labels, along with semi-supervised assumptions, can lead to better results than prior methods. For facial recognition, search queries to web repositories can be used as a source of unlabeled face image examples. The search engine then acts as a kind of classifier or labeler. However, the labeler is quite often incorrect. This thesis intends to show how proper handling of this information, combined with sparse kernel methods, can yield an improvement in face recognition performance.

Probabilistic sparse kernel methods are also an objective of this research. This thesis also intends to show how priors can be used to construct sparse kernel binary classifiers. Since they are probabilistic methods, they can be extended to other, more complex, models.

For facial expression recognition, instead of generating weakly labeled data, identity labels can be used as a kind of weak information. In this case, the assumption is that identity data can be used to construct representations for facial expression analysis in order to improve results. This research shows that this approach to additional information improves results for many expression tasks. In both cases, the methods discussed here obtain state-of-the-art results.

1.2.3 Hypotheses

Hypothesis 1 : Sparse kernel methods can be constructed in a way that allows them to be used as building blocks for other probabilistic methods.

Hypothesis 2 : The use of weakly labeled data can improve face recognition.

Hypothesis 3 : Identity information can and should be used to improve facial expression recognition.

1.2.4 Specific Contributions

This subsection presents the specific contributions that resulted from investigating these hypotheses.

- Novel probabilistic sparse kernel methods for binary classification and structured prediction.
- Construction of a weakly-labeled dataset from Youtube videos for face recognition.

- A method for learning classifiers using weakly-labeled data.
- Representations for expression recognition which normalize for identity.
- A method for improving facial expression recognition and performance driven animation using this representation.
- A method for improving key-point localization using identity normalization.

1.3 Organization of the Thesis

Chapter 2 provides a literature review of the topics addressed in this thesis, including a discussion of the state of the art in facial recognition and facial expression recognition. Also reviewed is work on graphical models, sparse priors, and semi-supervised learning, all of which are related to the methods used in the following chapters.

Chapter 3 is the first major theme of this thesis, which also provides many of the mathematical foundations required for the following chapters. In this chapter, sparsity and sparse priors are introduced and discussed in detail. Kernel methods are addressed as well. This chapter deals with the question of how to incorporate sparsity and kernel methods in a way which allows them to be extended to more complex models.

Chapter 4 is the second major theme of the thesis, applying the concepts from Chapter 2 to facial recognition with weak labels. Since unlabeled data is easily available, this chapter deals with the question of how to effectively use information associated with unlabeled data. That is, unlabeled data sometimes have natural "weak" labels that can be incorporated into learning. A novel method for weakly-labeled learning is fully developed.

Chapter 5 is the third and final major theme of the work, in which a probabilistic approach is used to "normalize" for identity. That is, representations for expression recognition often are subject to identity variation. This chapter introduces the idea of normalizing for identity by removing this variation. Since identity labels are often provided, this chapter deals with the question of how to effectively use identity information in order to improve facial expression recognition, performance-driven animation and facial key-point localization.

Chapter 6 concludes the thesis, summarizing the work presented in the thesis. The conclusion also suggests directions for further research.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a review of many of the following chapters. First a review of face recognition literature is presented, with particular focus on face recognition in images obtained from common sources such as the web and consumer photography. Then the literature of facial expression recognition, performance-driven animation and key-point localization are reviewed.

The machine learning methods used in the remaining chapters are reviewed as well in this chapter. In particular, the literature review focuses on kernel logistic regression and semi-supervised learning relevant to Chapter 3 and 4, as well as on factor analysis for chapter 5.

2.1 Face Recognition

Face recognition has a long history in artificial intelligence research. Many approaches and methods have been devised in order to help solve the problem. In earlier work, face recognition was focused on images in which variation was controlled to a large extent. That is, the facial images were collected in such a way to control for expression, pose, occlusion, background and other sources of variation. The field has moved toward more challenging data – images of faces captured in widely varying settings. The literature review is divided into two sections, first giving a brief overview of early work in Controlled Face Recognition and more recent work in Uncontrolled Face Recognition.

2.1.1 Controlled Face Recognition

Among the most widely cited facial recognition systems in the literature are those based on Principal Component Analysis (PCA) of intensity images, better known as eigen-faces, first presented by Sirovich et Kirby (1987) and used for recognition by Turk et Pentland (1991). In this method, test images are projected to a "face space" spanned by the eigenvectors corresponding to the largest eigenvalues from the singular value decomposition of a training set. Turk et Pentland (1991) based classification on the Euclidean distance of an example projected into a the respective face spaces of particular classes. This general subspace method has since been extended using discriminant analysis (Fisher's Linear Discriminant) by Belhumeur *et al.* (1996) and other types of factor analysis (*e.g.*, ICA, NMF, Probabilistic

PCA) (Bartlett *et al.*, 2002), (Guillamet et Vitria, 2002), (Moghaddam et Pentland, 1997)). Goel *et al.*. performed useful experiments with random projection which offer evidence that dimensionality reduction is a key task in facial recognition (Goel et of Computer Science University of Nevada, 2004).

Non-linear representations such as Locality Preserving Projection (LPP), a linear graph embedding method by Seung (2000) and later by Xu *et al.* (2010) and Sparsity Preserving Projections (SPP) ((Qiao *et al.*, 2010)), used for locally linear manifold representations has widely been applied to face recognition. Moghaddam *et al.* (2000) showed that similarity metrics, used commonly in Nearest Neighbor methods (NN), have also shown good performance. More recently, Wang *et al.*used a subspace created by merging KD-tree leaf partitions based on distance metrics and classification rates in order to improve facial recognition via LDA (Wang *et al.*, 2011). However, most of these methods have been applied to controlled databases. Yang et Huang (1994) describes similar face detection issues with complex background variation. Zhang et Gao (2009) provides an excellent review of the problems that arise due to uncontrolled pose variation.

In earlier work, facial recognition was dominated by these subspace methods, which have generally reported excellent accuracy. However these systems, termed "holistic approaches" in Zhao *et al.* (2003), require either an infeasible number of training examples to determine a reasonable set of basis vectors in the presence of wide variation or intensive preprocessing (*e.g.*pose alignment, background subtraction) to remove these sources of noise.

Not surprisingly, performance reported for the most commonly used image databases at the time – the AT&T ORL Face Database, MIT Face Database (Cambridge, 1994) Harvard and Yale Face databases, and the intensity image FERET database (Phillips *et al.*, 1998) – which consist of images in which sources of variation typically seen in natural images are highly controlled. Even the latter color FERET and FRGC face databases, which attempt to address this issue with the introduction of sources of variation, are also highly controlled, by natural image standards.

Figure 2.1 Example of "controlled" data, from Cambridge (1994).

In order to overcome some of these short-comings, the subspace methods were also extended to so termed "geometric" features (Zhao *et al.*, 2003), which attempt to build face descriptions or representations around key points or landmarks, such as the eyes, corners of mouth and nose (Yuille, 1991), or learned landmarks (Brunelli et Poggio, 1993). Early work in integration focused on modular subspace methods ("eigen-noses") by (Pentland *et al.*, 1994a), the incorporation of topological constraints (Local Feature Analysis) (Penev et Atick, 1996)

and more general network or graph-based approaches (Lades *et al.*, 1993).

Figure 2.2 Example of keypoints, from Wiskott *et al.* (1997)

Of these latter set of algorithms, the most widely cited are Active Appearance and Shape Models (AAM, ASM) (Cootes *et al.*, 1995b), (Cootes *et al.*, 2001)) and Elastic Bunch Graph Matching (EBGM) presented in Wiskott *et al.* (1997). These methods will be reviewed in more detail in Chapter 5.

However, the main drawback of these methods is the requirement of a labeling of the key points, and although a few databases exist, (FaceTracer (), BIOID (AG, 2001), PUT (Nordstrom *et al.*, 2004) among others), labeled examples are unsurprisingly difficult to obtain. Moreover, graph-based matching remains computationally intensive. In these cases good initialization of shape models and preprocessing (removing outliers and noise) are paramount. Again, however, training datasets are usually heavily controlled.

2.1.2 Unconstrained Face Recognition

Figure 2.3 LFW dataset : Examples of the 250x250 images, varying illumination, pose, occlusion.

As a response to the growing interest in less constrained labeled data the Labeled Faces in the Wild Dataset (LFW) was created by Huang *et al.* (2007b) in order to provide a far more natural composition of face images. Briefly, the database contains 13,233 color images of 5,749 subjects obtained by query from Yahoo News (Berg *et al.*, 2004). A query to the search engine was used to generate potential matches, and after face detection using the Viola Jones

face detector was applied, filtered by hand labeling the images. Each example is 250x250 and includes a region of 2.2 times the size of the bounding box recovered by the face detector or black pixel padding to reach the desired ratio and subsequently rescaled to attain the uniform resolution As a result, faces are generally in the center of the image and because the Viola Jones face detector was trained using frontal views only, the dominant pose is frontal.

Despite this, as depicted in Figure (2.3), the LFW database when compared with Figure (2.1), which contain examples from the much older AT&T Cambridge ORL database (Cambridge, 1994) presents a far more challenging, yet more realistic setting for experimental research. Even this small set of examples, depicted in Figure (2.3), exhibit, among other issues, indoor and outdoor illumination, occlusion, varying pose and even the complication of artificial makeup.

The database presents a challenging but realistic task. The task is made even more challenging by the distribution of the number of examples per subject. 99% of the subjects have less than 20 examples for both training and testing, 70% have only a single example.

While the mean of the number of training images per subject is a little more than 2, the fact that 70% of the data cannot be used for traditional train and test supervised classification makes facial recognition in the sense defined in Chapter 1, a difficult task. As such the stated focus is on face verification – pair matching – which roughly shares the same objective (Huang *et al.*, 2007b).

In verification, the goal is to distinguish if a given pair of images are faces belonging to a single subject. Another related sub-goal is one-shot learning, in which a classifier is built using at most one positively learned training example.

The University of Massachusetts at Amherst maintains an excellent summary page which also tracks best results (alf, 2010). Since the introduction of the database, the accuracy of pair matching has increased significantly, as seen in Refer to (alf, 2010) for a summary of results on the unrestricted task, which have slightly better accuracy.

Much of the work with the LFW database can be seen as experimentation with the specific subtasks inherent in pair-matching, which can be described as preprocessing feature selection, similarity computation, and classification. Preprocessing is an important task, including face localization, alignment and contrast normalization. Work by (Huang *et al.*, 2007a), (Huang *et al.*, 2008), (Taigman *et al.*, 2009) touches on this subject. After preprocessing, matching usually requires a choice of feature representation, as intensity information alone is too noisy for this data. Descriptor features used commonly in object recognition have been tested against the database, such as Haar-like features (Huang *et al.*, 2008), SIFT (Sanderson et Lovell, 2009), (Nowak et Jurie, 2007), LBP (Guillaumin *et al.*, 2009), (Wolf *et al.*, 2009a), (Wolf *et al.*, 2008a) and V1-like features (Gabor filter responses) (Pinto *et al.*, 2009). Finally,

comparison operators are commonly used to classify a test pair by thresholding to obtain a binary label (*i.e.*"same" or "different".) (Huang *et al.*, 2007a), (Huang *et al.*, 2008), (Nowak et Jurie, 2007), (Guillaumin *et al.*, 2009), (Sanderson et Lovell, 2009). The threshold is typically learned by SVM, (Pinto *et al.*, 2009), (Kumar *et al.*, 2009), (Wolf *et al.*, 2008a), (Pinto *et al.*, 2009), (Taigman *et al.*, 2009). As such, each subtask can be thought of as an exploration of entire avenues of research, and some work focuses more or less on a particular subtask.

At the introduction of the database, two baseline results were given, one using thresholded Euclidean distances between eigenfaces of test pairs and the other using a method based on clustering presented by Nowak and Jurie (Nowak et Jurie, 2007).

Alignment is usually an important preprocessing step in vision tasks, and (Huang *et al.*, 2007a), (Huang *et al.*, 2008), (Wolf *et al.*, 2009b) among others indicate that this is especially true for the LFW database. The literature includes three separate alignment methods, the unsupervised alignment (congealing and funneling) (Huang *et al.*, 2007a), a MERL procedure similar to a *supervised* alignment (Huang *et al.*, 2008), and commercial tool alignment (Wolf *et al.*, 2009b). Of these, the best results have been obtained with the commercial alignment tool, which is unfortunate due to the closed nature of commercial products. This serves to illustrates the importance of alignment.

Feature selection is classically an important step in any machine learning task. In general the work with LFW has utilized common feature transforms. The scale invariant feature transform (SIFT) (Lowe, 1999) has become one of the most popular feature representations for general computer vision tasks. Local Binary Patterns (LBPs) have also become popular, especially for facial recognition task (Ojala *et al.*, 2002a). Both are histogram representations computed over image regions which are used often in object recognition for robustness. SIFT features appear to have a degree of scale invariance and LBPs a measure of illumination invariance.

Some results seem to indicate that LBP's work better for face recognition tasks (Javier *et al.*, 2009). Feature learning has also been attempted using code-booking low-level features with trees (Nowak et Jurie, 2007), Gaussian mixtures (Sanderson et Lovell, 2009), and other means, such as the Linear Embedding descriptor of (Cao *et al.*, 2010). Currently best results without additional data have employed a descriptor-based approach built on landmark images and code-booking these patterns (Cao *et al.*, 2010). Multi-resolution LBP's have also obtained state of the art results (Wolf *et al.*, 2008a), while a concatenation of LBP's, SIFT, Gabor filter responses, and multi-resolutions LBP's offer state of the art results (Wolf *et al.*, 2009b). Good results using very simple features, *i.e.*pixel intensity histograms, have shown good results when combined with a learned similarity function, indicating that feature selection and similarity functions are complementary tasks (Pinto *et al.*, 2009).

Much work has been focused on the similarity functions. This is a quite general problem and many approaches are available in the literature. One obvious choice is Euclidean distance between two feature vectors (Huang *et al.*, 2007b), (Huang *et al.*, 2007a), and (Huang *et al.*, 2008). As implemented in a kernel function, distance formulation is fundamental to the performance of the SVM. A linear or Euclidean distance kernel corresponds to a Euclidean distance metric in the feature space.

One avenue of research is in learning a similarity function, typically through the optimization of a linear operator $\mathbf{A}$ which maximizes the similarity metric $(\mathbf{u}-\mathbf{v})^T\mathbf{A}(\mathbf{u}-\mathbf{v})$ – also known as the Mehalanobis distance. This procedure is also known as metric learning. Two methods for metric learning have been used on the LFW database, Information-Theoretic Metric Learning (ITML) (Davis *et al.*, 2007), (Kulis *et al.*, 2009), and Logistic Discriminant-based Metric Learning (LDML) (Guillaumin *et al.*, 2009). ITML can be described as minimizing a KL divergence between two multivariate Gaussians parameterized by $\mathbf{A}$ and $\mathbf{A}_0$ (typically $\mathbf{I}$), subject the constraint that the distances between corresponding labels be close, and vice-versa for differing labels. LDML, on the other hand treats the problem of finding $\mathbf{A}$ as parameterizing the probability of the label "same" or "different" according to a logistic regression model. LDML combined with an interesting nearest-neighbor algorithm Marginalized kNN (Guillaumin *et al.*, 2009) as well as ITML combined with so called "One-shot" scores (Taigman *et al.*, 2009) both give excellent results. The "One-shot" score is a LDA projection learned from a training set of a single positive example and a random large set of negative examples.

To date the best results have been obtained by integrating other sources of information, specifically outputs of attribute and component classifiers learned on different image databases and then applied to low level features (Kumar *et al.*, 2009). An enormous amount of data is effectively summarized by trained classifiers, and used at test time to compute a similarity function.

Pair matching, however, is not an identical problem as the original face recognition problem as defined in this thesis. Although a pair matching algorithm can be used to match each of the individual images previously marked or perhaps used in a kernel machine the standard approaches of multi-class classification can also be brought to bear. The work by (Wolf *et al.*, 2009a) and (Wolf *et al.*, 2008a) are the most representative. Wolf *et al.* (2008a) specifically addresses the question of how well descriptor-based methods used for pair-matching work for recognition tasks.

Descriptor based representations, often histograms of features, are popular in object recognition primarily based on their performance but also on simplicity. Popular descriptors such as SIFT (Lowe, 1999), Histograms of Oriented Gradients (HoG) (Dalal et Triggs, 2005)

and LBP (Ojala *et al.*, 2002b) have in common histogram representations as fixed length vectors each dimension of which contains a count of filter-like responses.

As histograms, these descriptors have a discrete density estimate interpretation. Like templates, however, they are easily interpretable. In challenges such as the Pascal Visual Object Classes Challenge (avo, 2010), combined with the SVM, these descriptors have proven quite effective.

Wolf *et al.* (2009a) and Wolf *et al.* (2008a) use LBP descriptors, which, as mentioned, are composed of histograms of filter-like responses. Unlike image gradients or Haar-like features, the LBP feature is quite different from a filter or convolution. Instead, the responses are discrete. They are designed as a coding of patches. Each patch is coded by assigning a binary number to each of its pixels based on the relationship of the central pixel to its neighbors. For a patch-size of 9 pixels, or 8 neighbors, a pixel is assigned 1 if the intensity of the pixel is greater than the central pixel, 0 otherwise. The resulting 8 binary numbers are called a pattern, and oriented from the top left-most pixel represents a discrete coding of the patch. This can be thought of as a filter response to one of two hundred and fifty-five "filters" in the eight neighbor case. The patterns are collected into histograms of size 255 (although the number is arbitrary as some authors have discussed "uniform" patterns, see (Ojala *et al.*, 2002b) for detail.) These descriptors capture much of the information available in gradient orientations while losing some gradient magnitude information but gain in illumination invariance. The descriptors can be computed in linear time and are simple to interpret.

The histograms are computed over non-overlapping regions of an image and concatenated into a single vector of binary pattern counts. Wolf *et al.* (2008a) additionally adds two LBP variants to the descriptor, a Three-Patch LPB (TPLBP) and Four-Patch LBP (FPLBP) descriptor, which attempt to address resolution variance by comparing patches instead of pixels. In the Three-Patch LBP, a central patch is compared with neighboring patches of size w using the L_1 difference between the pixel intensities in each patch. The pairs are chosen along a ring r pixels in radius and α patches apart, resulting in S patch pairings. Each S patch pairing corresponds to a single binary number in a S-vector which is 1 if the difference in L_1 distance between the central patch and the two pairs falls above a threshold or 0 otherwise. Depending on r and α, S can be of arbitrary size. In Wolf *et al.* (2008a), the patches are separated by one patch ($\alpha = 2$) and S is 8, corresponding to a radius of approximately 3, although Wolf *et al.* (2008a) admits to avoiding interpolation for efficiency. That is, a TPLBP of size 8 using a patch distance of size 2 with 3×3 patches is computed over a 9×9 patch centered around the central pixel of each patch. Once these values are computed, they are also histogramed over non-overlapping regions and concatenated into a single vector. The FPLBP, as the name implies, compares four patches, or more specifically

two pairs of patches. The two pairs are symmetric around the central pixel, again with each pair corresponding to a single bit in an vector, a thresholded difference in L_1 distance between the two pairs. An α parameter, set to 1 in Wolf *et al.* (2008a), determines an offset in pairs. Using 8 neighboring patches results in 8 pairs of patches, and 4 FPLBP patterns (8/2).

The combination of all three descriptors when combined with SVM and a novel algorithm called the One-Shot Similarity (OSS) kernel, resulted in best performance to date in the pair matching task, 76.53% recognition rate. However, they also published results on the recognition task, in which the a standard linear SVM was used with a varying number of classes. Not surprisingly, recognition rates were extremely good when a large number of labels were available, yielding $\sim 80\%$ accuracy in those cases. However, when the number of classes were relatively high at 100 the performance dropped below 50%.

In these experiments, the relationship of the number of classes to recognition rates are somewhat occluded by the issue of the number of training examples per class. Another set of experiments shows that recognition rates increase dramatically as the number of training images increase (Wolf *et al.*, 2009a). From this work it is implied that given a large enough number of labeled training examples a simple descriptor-based method should do quite well. (Wolf *et al.*, 2008a) also show that it is not necessarily the case that methods which improve pair matching lead automatically to improved results in recognition. Instead, the paper implies that the recognition task should be focused on obtaining more training examples, while the pair matching task should be focused on learning better distance metrics.

This underlies the desire to seek to use unlabeled data to improve recognition performance. In particular, the question is whether a large amount of weakly labeled faces can help replace the need for a large set of labeled faces.

2.2 Facial Expression Recognition

Expression variation is a major cause of difficulty in face recognition. Non-rigid deformations of the face can have dramatic changes in the appearance of a face. Despite this, human beings seem to be able to distinguish between these two sources of change fairly easily. Facial expression itself is a quantity that is of interest in a wide array of applications. For this reason, the ability to automate facial expression recognition has received a great deal of attention. Humans communicate emotions and other information through facial expression among other non-verbal cues. Automatic recognition emotional states could have wide-ranging applications in human computer interaction, medical diagnosis and entertainment. However, although a great deal of progress has been made, recognition of facial expressions remains a challenging task.

Expression recognition has been heavily influenced from work on face recognition. Typically, systems for expression recognition tend to follow a pipeline similar to face recognition, pre-processing, feature extraction and classification.

2.2.1 Expression Parameterization

As mentioned in the Introduction, facial expression can be interpreted in more than one way. In natural language, it is often interpreted as the deformation of the face associated with activations of facial muscles. The prototypic expressions, which include "Anger," "Disgust," "Fear," "Happiness," "Sadness," and "Surprise," produce a natural set of facial unit activations (Ekman et Rosenberg, 2005). Because the activations are spontaneous and distinct, the emotional state can be used synonymously with the muscle activations which lead to the facial expression. Because of this, an important class of expression parameterizations are those based on facial Action Units (AU) (Kanade *et al.*, 2000), usually associated with Facial Action Coding System (FACS) developed by Ekman in the 1970's (Ekman et Rosenberg, 2005).

FACS was developed in order to code for facial expressions in behavioral psychology experiments. It is the best known and most widely used coding system. FACS are based on 44 AUs, which are observable and recognizable face deformations based on muscle groupings, *e.g.*1 = inner brow raiser, 2 = outer brow raiser, etc. Using a reference, a human coder is tasked to label each FACS AU in an image. These represent particular expressions. In Figure (2.4), action units 1, 2, 5 and 27 are represented in the sequence.

Figure 2.4 Example of a sequence of FACS coded data from Cohn-Kanade dataset (Kanade *et al.*, 2000), which contains two views of the sequence. The AU of the final frame is coded as 1+2+5+27, Inner Brow Raiser+Outer Brow Raiser+Upper Lid Raiser+Mouth Stretch

FACS AU detection has become an intensive area of research as well, due to the recognition that recognizing emotional states from image data relies on effective parameterization.

An important issue with FACS AU detection is the combinatorial nature of the AU descriptor space. Treating each of AU as a discrete label is unreliable because of the complexity of human face expressions ; combinations of different AUs can create non-rigid deformations which cannot always be considered independent given the set of AU activations. In order to treat these as dependent or confounding variables would require a much higher number of labeled examples – the so-called "curse of dimensionality." For this reason, most AU detection research limits the set of AU activations to only those that correspond to an semantically coherent expression, such as "smile" or "frown," induced by one of the prototypical expressions.

Much like face recognition, work with FACS has been limited by the high cost of labeling. Earlier work with databases such as the CMU-Pittsburgh AU-Coded Face Expression Image Database (Kanade *et al.*, 2000), for example, consists of only 300 images of frontal face subjects. Larger datasets such as the CMU PIE Database provide limited labels (Sim *et al.*, 2002). For a review of early databases, see (Pantic *et al.*, 2005).

The Extended Cohn Kanade (CK+) database (Lucey *et al.*, 2010) for emotion recognition and facial action unit tasks is a more recent database. The CK+ dataset consists of 593 image sequences from 123 subjects ranging in age from 18 to 50, 69% of whom are female and 13% of whom are black. The images are frontal images of posed subjects taken from video sequences. Each sequence contains a subject posing a single facial expression starting from a neutral position. The sequence consists of sampled frames from video in which the final posed position is labeled with FACS action units. In addition, emotion labels, consisting of the expressions seven prototypic expressions, which includes "Contempt," are provided for 327 of the 593 sequences. However, because only the first and last image are labeled with FACS AUs, the size of the labeled set containing non-neutral expressions is 327 images.

2.2.2 Features for Expression Recognition

To date, a wide variety of methods have been used as feature representations for expression recognition. Many of these are based on feature representations drawn from face recognition and object recognition research.

These include Gabor Wavelets (Lyons *et al.*, 1999), (Bartlett *et al.*, 2004), (Tong *et al.*, 2007), key-point locations (Zhang *et al.*, 1998), (Tian *et al.*, 2001), (Valstar *et al.*, 2005), Local Binary Patterns (Shan *et al.*, 2009), multi-view representations (Pantic et Rothkrantz, 2004), and optical flow-based approaches (Essa et Pentland, 1997).

(Donato *et al.*, 1999) compares several of these methods for AU detection, including holistic approaches such as PCA, LDA and ICA in addition to optical flow. They conclude that best performance on their data was achieved using Gabor wavelets and ICA. Recognizing the difficulty of expression recognition from 2D images alone, many approaches to expression recognition use other modalities, principally video sources, multi-view sources and 3D geometry.

Video sources offer the possibility to detect temporal change in expressions using optical flow (Black et Yacoob, 1997), (Otsuka et Ohya, 1997). In essence, this turns the problem into one of tracking. That is, the goal is track changes in facial images and map them to the corresponding expression. (Essa et Pentland, 1997), (Li *et al.*, 1993) and (Terzopoulos et Waters, 1993) combine this sequence data with 3D geometry, using a face mesh models to help model the face physiology. (Terzopoulos et Waters, 1993) tracks active contour snakes

around fiducial points and parameterizes expressions as key-point locations and their time derivatives. (Essa et Pentland, 1997), meanwhile, tracks changes in global optical flow and uses this information to map changes in the 2D image. They accomplish this by mapping a neutral pose 2D image onto a sphere and tracking optical flow to model deformations in the mesh. Hidden Markov Models (HMM) have also been applied to expression sequence data (Otsuka et Ohya, 1997). Sequence data suggests that expressions have a time component that may be necessary for discrimination.

The addition of 3D data is often necessary when attempting to discriminate AUs which may be out-of-plane. An example used in (Sandbach *et al.*, 2012) is a lip-pucker. From a frontal view in 2D, the lip-pucker would be difficult to detect. There is a large body of work using 3D data to recognize expressions. A common approach is using structured light, dense point correspondence and a manifold learning technique to learn an expression recognition system (Chang *et al.*, 2005), (Wang *et al.*, 2004b), (Tsalakanidou et Malassiotis, 2010). (Sibbing *et al.*, 2011) use 3D Morphable Models (3DMM) instead of point correspondence. Since this thesis is concerned primarily with 2D expression recognition, for a comprehensive review of 3D techniques see (Sandbach *et al.*, 2012).

As in face recognition, an assortment of discriminative methods have been shown to provide good performance. Neural nets are used in (Bartlett *et al.*, 2004), (Fasel, 2002), (Tian *et al.*, 2001), and (Neagoe et Ciotec, 2011). SVMs are used in (Valstar *et al.*, 2005), (Kotsia et Pitas, 2007) and (Valstar *et al.*, 2011). Bayesian networks have also been used to classify expressions (Cohen *et al.*, 2003), (Lien *et al.*, 1998), (Tong *et al.*, 2007). Boosting approaches have also been applied (Zhu *et al.*, 2009). This list is not comprehensive, for a review of these techniques, see (Zhao *et al.*, 2003) and (Zeng *et al.*, 2009).

2.2.3 Subject Independent Emotion Recognition

One important consideration is whether the emotion recognition system is robust to identity. While pose and illumination changes have been studied in detail for face recognition, an important source of variation for expression recognition is identity. Many expression recognition systems are evaluated on single images of expression without carefully removing all images of test identities from the training set.

A subclass of work in emotion recognition is focused on this issue. Many of the approaches to identity invariance are based on multi-linear analysis (Tenenbaum et Freeman, 2000), (Vasilescu et Terzopoulos, 2002). Multi-linear analysis is a type of factor analysis in which factors modulate each other contributions multiplicatively. However, when all other factors are held constant, the interaction of the factor of variation is linear.

Bilinear analysis was first applied by (Hongcheng Wang et Ahuja, 2003), in which ex-

pression and identity were used as the two factors. The algorithm for separating the two modes is based on higher-order singular value decomposition (HOSVD) or sometimes called N-SVD, which generalizes SVD to higher order tensors. In HOSVD, training data must be aligned (each training identity must have corresponding expression data) so that the tensor is full rank. (Hongcheng Wang et Ahuja, 2003) used gray-scale pixel intensity images, but HOSVD for expression recognition has been applied to sequences (Lee et Elgammal, 2005) and key-point locations (Abboud et Davoine, 2004), (Cheon et Kim, 2009) as well. Bilinear factorization for expression recognition have also been applied to 3D data (Mpiperis *et al.*, 2008). The method can be interpreted as a preprocessing step, as the expression sub-space coordinates are used as features, rather than features derived from the raw input. (Tan *et al.*, 2009) used HOSVD to build similarity-weighted features. HOSVD can also be used for higher order tensors, combining pose or illumination as well (Zhu et Ji, 2006). Since the model is generative, expressions may also be synthesized or transferred (Cheon et Kim, 2009), which can be useful for performance-driven animation. Multi linear analysis has also been applied to key-point tracking using the AAM. However, one issue with HOSVD is the requirement that the data be aligned so that each entry of the image tensor does not have missing data.

Alternatives to multi-linear analysis include using more complex non-linear modeling approaches, as the probability distributions of the expressions are presumed to be at least multi-modal, containing a mode for each identity. One natural approach is the use of mixture models to model expression probabilities, (Liu *et al.*, 2008) and (Metallinou *et al.*, 2008). In this case, identities are assumed to form clusters of expressions. However, this does not directly correspond to the identities, as the unsupervised models do not make use of the identity labels.

A more direct approach is pursued by Fasel (2002), in which convolutional neural nets (CNN), are structured to both predict both identity and expression. Fasel (2002) note that when both are modeled and predicted simultaneously, the performance of the expression classifier increases. Another approach using CNN's is used in Matsugu *et al.* (2003), where a modular system combines a neutral face with a training image to learn features used to predict emotion labels. This is similar to direct identity normalization in Weifeng Liu et Yan-jiang Wang (2008), which used the difference in LBP histograms between an image of the neutral expression and training images to learn "identity-normalized" features. This is an interesting approach in that it effectively uses a learned feature representation before classification training. Neagoe et Ciotec (2011) also use this approach, by using PCA for dimensionality reduction including the testing data as well. Interestingly, Neagoe et Ciotec (2011) showed good performance when adding "virtual" examples to the training set, indicating that a lack of labeled training data appears to be a problem.

In general, systems for expression recognition should take into account identity, especially for databases of 2D frontal face images. Results using multi-linear analysis and other subject-independent studies have shown improved results on facial expression recognition. The additional information from the identity label appears to be useful for learning expressions. Multi-linear approaches, moreover, allow for extension to generation and synthesis. They can also easily be generalized to other labeled sources of variation. In general, however, these systems require a rigid alignment of data (in the form of a full tensor). This thesis is designed to show an alternative to multi-linear analysis which can be used without this requirement which can also be applied to performance driven animation and key-point tracking easily. This alternative is based on probabilistic modeling, which is the subject of the next section.

2.3 Graphical Models in Machine Learning

The basis for the models used throughout the following chapters are grounded in the following sections. The first part briefly reviews some basic concepts used in this thesis. Graphical models are used throughout this thesis and are first reviewed briefly here. This provides an introduction to kernel logistic regression and principal components analysis (PCA) then to semi-supervised learning, weakly labeled learning and then finally to feature-learning methods.

2.3.1 Graphical Models

Machine learning is an extremely diverse field and to review the literature completely is beyond the scope of this thesis. However, a central tenet in this thesis is that probabilities should be used to model complex problems. This is because by using probabilities, formulating complex models and solving for quantities of interest can be accomplished by following the relatively simple rules of probability. However, to describe such models using algebraic notation can be cumbersome. This has lead to the widespread use of graphical modeling, in which probability distributions are specified and visualized using graphs (Pearl, 1988), (Whittaker, 1990). For a more thorough examination of learning and inference in general graphical models, see Jordan (2004), Bishop *et al.* (2006),Whittaker (2009),Koller et Friedman (2009), among others.

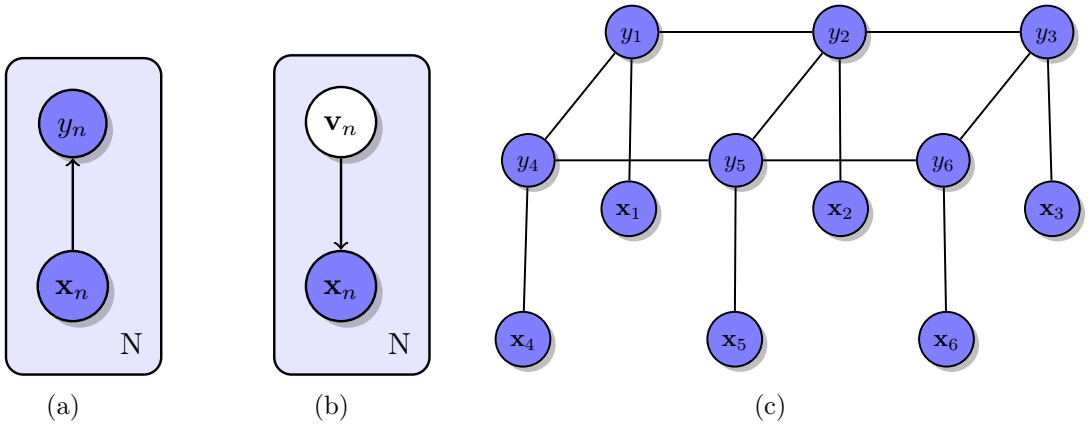


Figure 2.5 Example graphical models : 2.5(a) visualizes logistic regression, used in Chapter 3 and Chapter 4. 2.5(b) visualizes Principal Components Analysis, discussed in Chapter 5. 2.5(c) visualizes a lattice structured CRF, developed further in Chapter 3.

A graphical model consists of a set of nodes, which specify random variables, and edges which specify probabilistic relationships between these variables. Figure (2.3.1) presents three separate graphical models. The first graphical model, Figure (2.5(a)) describes a model in which a variable $\mathbf{x}_n$ and a variable y_n are connected by a directed edge. The "plate notation," signified by the pale blue node surrounding the two variables indicates the presence of a set of N independent pairs of variables (Spiegelhalter et Lauritzen, 1990).

The directed edge signifies that the random variable y_n is conditionally dependent on the random variable $\mathbf{x}_n$. In other words, the graphical model depicts a set of N independent conditional distributions $p(y_n|\mathbf{x}_n)$. In this case, both the nodes y_n and $\mathbf{x}_n$ are shaded, indicating that they are observed. This means that the model variables are not hidden – the value of both random variables are known. Thus, the conditional distribution of the data is given by $\prod_n^N p(y_n|\mathbf{x}_n)$, by the product rule of probability.

If only the conditional distribution is sought, the model is called discriminative. This is because the model can only be used to determine the probability of a particular y given some input variable $\mathbf{x}$. In contrast, the model can also be trained to optimize the joint probability $\prod_n^N p(y_n, \mathbf{x}_n)$, which is given by $\prod_n^N p(y_n|\mathbf{x}_n)p(\mathbf{x}_n)$, provided $p(\mathbf{x}_n)$ is available. In this case, the model is called generative, because it allows for the generation of samples, by sampling an $\mathbf{x}$ from $p(\mathbf{x})$ and subsequently sampling y from $p(y|\mathbf{x})$. In this thesis, the discriminative model, which will be parameterized to form a particular procedure known as logistic regression, is developed further in both Chapters 3 and Chapter 4.

The second model, shown in Figure (2.5(b)), depicts a probability model in which not all random variables are observed. The model is visually similar to the first, but with two

main differences. First, the directed edge between the random variable $\mathbf{x}_n$ and the variable $\mathbf{v}_n$ indicates conditional dependence in the form $p(\mathbf{x}_n|\mathbf{v}_n)$. This suggests a generative model – the model should be learned by maximization based on the joint probability $\prod_n^N p(\mathbf{x}_n|\mathbf{v}_n)p(\mathbf{v}_n)$. This is also called a hidden or latent variable model since $\mathbf{v}_n$ is unobserved, as indicated by the lack of shading.

This model can be parameterized to form the basis for Probabilistic Principal Component Analysis (PPCA), as presented by Tipping and Bishop (Tipping et Bishop, 1999). PPCA serves as the basis for the model used in Chapter 5. The literature and background for this model and others based on it are discussed in following sections of this chapter.

The third model, also shown in Figure (2.5(c)), represents a more complicated setting in which a "lattice" structure governs the distribution of the variables. In this case, the edges are undirected. In this case, this signifies that the variables connected by an edge are dependent in some way. In other words, if two variables are connected by a path via a set of edges, they must be conditionally dependent. On the other hand, if all sets of paths from one variable to another are "blocked" by an observed node, then the variables are conditionally independent given the observed node. This is also true for any subset of nodes. Thus it is easy to determine conditional independence in an undirected graph. For example, in the lattice structure, shown in given any node y_i, $\mathbf{x}_i$ is independent of any other node in the graph. Therefore, any variable is conditionally independent from any other variable in the graph, given all the nodes to which it is directly connected. Any undirected graphical model can therefore be described in terms of maximal cliques, so that if g indexes the cliques, the probability of the undirected graph is given by

$$p(\mathbf{X}) = \frac{1}{Z}\prod_g \Psi_g(\mathbf{x}_g), \tag{2.1}$$

where $\mathbf{X}$ represents all the variables in the graph and g indexes the cliques. The normalization constant $\mathbf{Z}$ is given by

$$Z = \sum_{\mathbf{x}}\prod_g \phi_g(\mathbf{x}_g). \tag{2.2}$$

In this particular case, this is a lattice structured model, so the maximal cliques are the edges between each neighboring variable. There are two kinds of edges, those between the labels y_n and the data $\mathbf{x}_n$, and those between neighboring y_i, y_j. These can be represented by two kinds of potential functions, $\phi(y_i, y_j)$ and $\psi(y_i, \mathbf{x}_i)$, where g again indexes the cliques.

If the optimization is over the joint distribution $p(\mathbf{y}, \mathbf{X})$, *i.e.*, generatively, the model can be interpreted as a Markov Random Field (MRF) (Moussouris, 1974), (Kindermann *et al.*, 1980). If it is trained discriminatively (maximizing $p(\mathbf{y}|\mathbf{X})$), then it can be viewed as a Conditional Random Field (CRF), (Lafferty *et al.*, 2001), (Sutton et McCallum, 2010). The CRF model is adapted in Chapter 3, and so this model is reviewed in this Chapter.

2.3.2 Logistic Regression

Logistic regression is an approach to modeling the relationship between real valued input data $\mathbf{x}$ and a dependent binary y variable. It has a long history dating back to the 1800's, (Cramer, 2003), but popularized in the 1960's by Cox (Cox, 1958) and used for classification in machine learning (), (McFadden, 1984). It can be interpreted as a generalized linear model (GLM) or generalized additive model (GAM), in that it generalizes linear regression to binary dependent variables through the use of the logistic function (Nelder et Wedderburn, 1972), (Hastie et Tibshirani, 1995). A probabilistic interpretation is straight-forward. As seen in the previous section, the model describes a conditional probability. The data, defined as $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, where $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$, and $y = \{y_n\}_{n=1}^N$, and $\mathbf{x}_n \in \mathbb{R}^D$ and $y_n \in \{-1, 1\}$, are modeled by the conditional probability

$$p(\mathbf{y}|\mathbf{X}) = \prod_n^N p(y_n|\mathbf{x}_n). \tag{2.3}$$

Logistic regression is formulated from this model by letting

$$p(y_n|\mathbf{x}_n) = \sigma(y_n \mathbf{w}^T \mathbf{x}_n), \tag{2.4}$$

where σ is the logistic function

$$\sigma(a) = \frac{1}{1 + \exp(-a)}. \tag{2.5}$$

Now we can rewrite the conditional probability of the data as

$$\prod_n^N p(y_n|\mathbf{x}_n; \mathbf{w}) = \prod_n^N \sigma(y_n \mathbf{w}^T \mathbf{x}_n), \tag{2.6}$$

where $\mathbf{w}$ is a parameter vector of weights on the features $\mathbf{x}$. The connection to generalized linear models and to generalized additive models (GAM) (Hastie et Tibshirani, 1995), can be seen by the fact that the model is log-linear.

Optimizing the conditional distribution can therefore be accomplished by maximizing the log of Equation (2.6) with respect to $\mathbf{w}$

$$\max_{\mathbf{w}} \sum_{n}^{N} \log \sigma(y_n \mathbf{w}^T \mathbf{x}_n). \tag{2.7}$$

This is generally accomplished by gradient descent or second-order methods such as conjugate gradient (CG) (Hestenes et Stiefel, 1952), or iteratively re-weighted least squares (IRLS) (). The process of maximizing the log of the probability distribution is an example of maximum likelihood, or more precisely maximum conditional likelihood, learning for graphical models.

Once optimal model parameters $\mathbf{w}$ are obtained, given any new test data $\mathbf{x}$, logistic regression can be used for classification by taking $y = \text{argmax}_{\ y} p(y|\mathbf{x})$. Logistic regression has been employed extensively in machine learning and for vision. In chapters 3, this simple model will be extended to allow for additional constraints and other advantageous properties. In chapter 4, logistic regression forms the basis for the classifier which leverages weakly labeled data. For a more in-depth treatment of logistic regression see (Bishop *et al.*, 2006) among others.

2.3.3 Principal Component Analysis

Principal components analysis (PCA) is a commonly used machine learning method used for both pre-processing data as well as for dimensionality reduction. As described by Pearson and developed by Hotelling, PCA is defined as an orthogonal projection of the data to a linear subspace (Pearson, 1901), (Hotelling, 1933). In particular, this linear subspace is the one in which the variance of the data is maximized. A simpler way to understand PCA is as a rotation of the original axes of the data or change of coordinate system to one in which the variance of the data in the new space is maximized.

Let $\{\mathbf{x}\}_{\{n=1,2,...,N\}}$ represent a set of N data vectors such that $\mathbf{x} \in \mathbb{R}^D$. Let $\mathbf{S}$ be the sample covariance, and μ be the sample mean,

$$\mathbf{S} = \frac{1}{N} \sum_{n}^{N} (\mathbf{x}_n - \mu)(\mathbf{x}_n - \mu)^T, \tag{2.8}$$

then PCA finds a set of orthogonal vectors $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_M$, such that the projected variance, *i.e.*$\mathbf{u}_m^T \mathbf{S} \mathbf{u}_m$, is maximized. It can be shown that $\mathbf{u}_m$ is given by the m^{th} eigenvector of the covariance matrix $\mathbf{S}$, with eigenvalue λ_m. If $\mathbf{L}$ represents the $M \times M$ diagonal matrix with $\mathbf{L}_{m,m} = \lambda_m$ and $\mathbf{u}$ is the $D \times M$ matrix comprised of each $\mathbf{u}_m$ as the m^{th} column, then $\mathbf{SU}\mathbf{u} = \mathbf{U}\mathbf{u}\mathbf{L}$, by the eigen-decomposition.

The projection $\mathbf{U}\mathbf{u}^T(\mathbf{x}_n - \mu)$ gives a vector $\mathbf{v}_n$ which represents the transformed data. In this case, $\mathbf{v}_n$ has mean 0. Typically, $M < D$, so that because U is comprised of orthogonal columns, $\hat{x}_n = \mu + \mathbf{U}\mathbf{u}\mathbf{v}_n$ represents a method of compression.

In cases where $M = D$, the procedure can still be useful. The projection $\mathbf{v}_n = \mathbf{L}^{-\frac{1}{2}}\mathbf{U}\mathbf{u}^T(\mathbf{x}_n - \mu)$ results in the set $\{\mathbf{v}\}_{\{1,2,\ldots,N\}}$ having both mean 0 and a covariance close to identity. Use of this projection for preprocessing data is also called sphereing, since the projected data ($\mathbf{v}_n$) is expected to have equal unit variance and uncorrelated components. This can have beneficial effects for models which assume homoscedasticity, for example.

Although PCA was not formulated as a probabilistic method originally, graphical models can be used to describe and arrive at a probabilistic version of PCA (Tipping et Bishop, 1999). Again, the nodes represent the random variables $\mathbf{v}_n$, and $\mathbf{x}_n$. In contrast to the previous model, $\mathbf{v}_n$ is depicted as a hidden variable. The link between $\mathbf{x}_n$ and $\mathbf{v}_n$ indicate that the probability of the data is given as $p(\mathbf{x}_n|\mathbf{v}_n)$.

In the graphical model, as shown before, the optimization is of the joint probability is given by

$$\prod_n^N p(\mathbf{x}_n|\mathbf{v}_n)p(\mathbf{v}_n) \tag{2.9}$$

In PPCA $p(\mathbf{v}_n)$ is assumed to be zero-mean Gaussian with identity covariance, $\mathcal{N}(\mathbf{v}_n|0, \mathbf{I})$. The conditional probability is also Gaussian, with

$$p(\mathbf{x}_n|\mathbf{v}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{W}\mathbf{z}_n + \mu, \Sigma). \tag{2.10}$$

The covariance is assumed to be isotropic : $\Sigma = \sigma^2\mathbf{I}$, where σ^2 is a scalar. Since both $\mathbf{x}_n$ and $\mathbf{z}_n$ are Gaussian, the marginal distribution $p(\mathbf{v}_n|\mathbf{x}_n)$ are also Gaussian and can be expressed in simple forms. In particular, the marginal distributions $p(\mathbf{x}_n)$, obtained by integrating out $\mathbf{z}_n$, can be expressed as a Gaussian in terms of $\mathbf{W}, \mu$ and σ^2,

$$p(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n|\mu, \mathbf{W}\mathbf{W}^T + \Sigma). \tag{2.11}$$

Tipping and Bishop show that the maximum likelihood in this case is obtained when $\mathbf{W}$ can be decomposed in the matrices

$$\mathbf{W} = \mathbf{U}\mathbf{u}(\mathbf{L} - \Sigma)^{1/2}\mathbf{R}, \tag{2.12}$$

where $\mathbf{R}$ is an arbitrary orthogonal matrix, and

$$\sigma^2 = \frac{1}{D-M} \sum_{i=M+1}^{D} \lambda_i, \tag{2.13}$$

or the average of the remaining eigenvalues of the data covariance matrix. Therefore the maximum likelihood solution and sphereing is equivalent, up to an orthogonal $\mathbb{R}$, when $D = M$.

The graphical model, however, can lead to a diverse set of methods. An identical model with non-isotropic covariance, in particular diagonal covariance, leads to Factor Analysis (Thurstone, 1931), (Toutenburg, 1985). This model will be discussed further in Chapter 5, towards making use of additional information.

2.3.4 Conditional Random Fields

The Conditional Random Field is a form of structured prediction, as the output labels y_{ij} depend on each neighbor as well as the input $\mathbf{x}_{ij}$, presented by Lafferty *et al.*, (Lafferty *et al.*, 2001). The CRF is a discriminative model, as both the labels y and the data are observed, and the conditional distribution, $p(\mathbf{y}|\mathbf{X})$ is the optimization target. For a detailed tutorial, see (Sutton et McCallum, 2010).

According to the graph shown in Figure (2.5(c)), the cliques are defined as the pairwise connected nodes in the graph. The plate notation is omitted for simplicity, but in the usual case, many examples are available. Associated with each clique is a potential function. Let ϕ be the potential functions for the pairwise y cliques and let ψ be the potential functions for the pairwise x_{ij}, y_{ij} cliques. Let G_x be the index set for y, x cliques, and G_y be the index set for label cliques. As discussed previously, the conditional distribution for an example is then given by

$$p(\mathbf{y}|\mathbf{X}) = \frac{1}{Z} \prod_{g \in G_x} \psi(y_g, \mathbf{x}_g) \prod_{i,j \in G_y} \phi(y_i, y_j). \tag{2.14}$$

Again, Z is the normalization term

$$Z = \sum_{\mathbf{y}} \prod_{i \in G_x} \psi(y_i, \mathbf{x}_i) \prod_{i,j \in G_y} \phi(y_i, y_j). \tag{2.15}$$

Since it is easiest to work with log-linear functions, this is usually the case. Letting λ and γ be two parameter vectors,

$$\phi = \exp(-\lambda^T \rho_\lambda(y_i, \mathbf{x}_i)) \tag{2.16}$$

$$\psi = \exp(-\gamma^T \rho_\gamma(y_i, y_j)), \tag{2.17}$$

where ρ is a possibly vector-valued features of the combined variables. ρ is known as a feature function. For example, in an image segmentation application where each pixel is a node in a lattice, $\rho_\gamma(y_i, y_j)$ could be $\mathbf{1}_{\{y_i=y_j\}}$.[1] In that example, learning the weight γ can be interpreted as learning the importance of the neighboring pixels having the same label.

Therefore, the log likelihood can be expressed as

$$p(\mathbf{y}|\mathbf{X}) = \frac{1}{Z} \exp\Bigg(- \sum_{i \in G_x} \lambda^T \rho_\lambda(y_i, \mathbf{x}_i) - \sum_{i,j \in G_y} \gamma^T \rho_\gamma(y_i, y_j) \Bigg). \tag{2.18}$$

This can be optimized by gradient descent, by taking derivatives for the model parameters λ and γ. This can be easier to see in log form, with

$$\log p(\mathbf{y}|\mathbf{X}) = - \sum_i \lambda^T \rho_\lambda(y_i, \mathbf{x}_i) - \sum_{i,j} \gamma^T \rho_\gamma(y_i, y_j) - \log Z \tag{2.19}$$

so that the derivative for λ is given by

$$\nabla_\lambda \log p(\mathbf{y}|\mathbf{X}) = - \sum_i \rho_\lambda(y_i, \mathbf{x}_i) - \nabla_\lambda \log Z. \tag{2.20}$$

Because

$$\nabla_\lambda \log Z = - \sum_{\mathbf{y}} \Bigg(\sum_i \rho_\lambda(y_i, \mathbf{x}_i)) \frac{1}{Z} \exp(- \sum_{i=1}^{N} \lambda^T \rho_\lambda(y_i, \mathbf{x}_i) - \sum_{i,j \in G} \rho_\gamma(y_i, y_j; \gamma) \Bigg), \tag{2.21}$$

it is often expressed as the negative conditional expectation

$$-\nabla_\lambda \log Z = \mathbb{E}[\sum_i \rho_\lambda(y_i, \mathbf{x}_i)|\mathbf{X}], \tag{2.22}$$

1. $\mathbf{1}_{\{a\}}$ is the indicator function, 1 if a is true, and 0 otherwise.

and the gradient descent update (the negative gradient) is given by consists of

$$-\nabla_\lambda \log p(\mathbf{y}|\mathbf{X}) = \sum_i \rho_\lambda(y_i, \mathbf{x}_i) - \mathbb{E}[\sum_i \rho_\lambda(y_i, \mathbf{x}_i)|\mathbf{X}] \tag{2.23}$$

$$-\nabla_\gamma \log p(\mathbf{y}|\mathbf{X}) = \sum_{i,j} \rho_\gamma(y_i, y_j) - \mathbb{E}[\sum_i \rho_\gamma(y_i, y_j)|\mathbf{X}]. \tag{2.24}$$

Since the objective is maximized with the gradient at 0, this has the satisfying interpretation as preferring that the model expectation match the observed information. The main difficulty in this structure is the difficulty of computing the expectation, since the space of labels grows exponentially. There are a number of approximate ways to compute the marginal probabilities required to compute the expectation *e.g.*$\rho_\lambda(y_i, \mathbf{x}_i)$. This includes pseudo-likelihood training, loopy belief propagation (Pearl, 1988),(Murphy *et al.*, 1999), (Yedidia *et al.*, 2000), mean field (MF) and other variational methods (Winn *et al.*, 2005), and graph cuts (GC) (Kolmogorov et Zabih, 2004), (Boykov *et al.*, 2001). Full details of these methods are reviewed in more detail in seen in. More discussion of CRF's appear in Chapter 3.

In summary, graphical modeling is an important technique for probabilistic modeling. Using these graphs to specify distributions allows for the visualization of even very complex probability distributions. The choice of objective, *i.e.*conditional, joint or marginal, and parameterization gives rise to particular algorithms. In this section, three examples were given. The first gives rise to logistic regression, discussed in Chapters 3 and 4, the second to PCA, discussed in Chapter 5, and Conditional Random Fields, discussed in Chapter 3. In all three examples, data is assumed to be labeled or unlabeled. In this case, PCA treats the data as unlabeled, although $\vee$ can be considered as a type of label. Therefore PCA is an unsupervised method, whereas logistic regression and CRF are supervised approaches. However, labels are usually difficult to obtain, because they require manual effort. Large amounts of unlabeled data are usually easy to obtain. However, it seems difficult to learn a method for mapping data to labels without labeled examples.

The desire to produce better algorithms using both labeled and unlabeled data has resulted in a diverse field of research called semi-supervised learning (SSL). As with strictly supervised tasks, learning the relationship between the labels and the examples is the goal. However, in semi-supervised learning not all the examples are associated with labels as in unsupervised learning.

2.3.5 Semi-Supervised Learning

Generative models are the obvious choice for making use of unlabeled data. For example, Nigam *et al.*. implement a Naive Bayes Model using on a text classification task, labeled exampled are "clamped" to cluster components and EM is used to discover probable labelings for the unobserved labels (Nigam *et al.*, 2006). In the hidden variable interpretation, the missing labels are treated as hidden while available labels are treated simply as observed.

Similarly, Basu *et al.*apply label information to a purely generative task – clustering (Basu *et al.*, 2004). Here, the goal is to find a clustering in which the labels act as constraints on which examples should be or should not be clustered together. The additional constraints acts on the cluster distortion measure, which effects both labeled and unlabeled data. In both methods, the parameters of the underlying clusters are assumed to generate both the label and the features. Although it is difficult to evaluate generative tasks, both results show that unlabeled data help most where the number of labels is fairly small.

However, one issue with generative approaches is that part of model fit is determined by how well $p(\mathbf{x})$ is modeled. Since the marginal is not the quantity of interest, on principle it seems better if we do not have to model it. Approaches which can be called diagnostic, (Seeger, 2006), which hope to avoid this and model only the conditional distribution $p(y|\mathbf{x})$, have been applied in several settings. To use the unlabeled data, however, requires additional assumptions, which are, more or less, implicit in generative methods. One of these strong assumptions is low density separation, that is, decision boundaries should avoid passing through regions of high marginal density.

Transductive SVM (TSVM) is based on the extension of the large-margin classifier to the transductive setting (Joachims, 2006). That is, the size of the margin is determined by both the labeled and unlabeled data, typically the test data. This requires a labeling of the unlabeled data and such labelings are restricted to be consistent with the classifier learned on labeled data. The final hyperplane is chosen based on maximizing the margin in the combined space of the labeled and pseudo-labeled data. This approach is appealing because it explicitly encodes a low density separation constraint, that is decision boundaries are located where $p(\mathbf{x})$ is low, and the margin ensures that the decision boundary is passes optimally through this region (Joachims, 2006). This is called transductive because rather than learning a function inductively and then evaluating outputs on a test set, the test set itself is used for learning. The main drawback with the TSVM is that the number of pseudo-labelings is combinatorial. Although relaxations exist, greedy search-based approximate methods still appear to yield the most scalable algorithms.

In probablistic terms, low density separation assumptions can be implemented as a constraint. In terms of optimization, this can also be applied as a regularization penalty. A probabilistic

low density separation algorithm is given by Grandvalet and Bengio (Grandvalet et Bengio, 2004) in which a logistic regression objective is combined with a penalty on the entropy of the conditional distribution $p(y|\mathbf{x}, h = 1; \theta)$, where $h = 1$ denotes that the label is missing. A similar penalty presented by Corduneanu and Jaakola involves minimizing the difference in mutual information between the label and the example $\mathbf{x}$, and the region(s) from which $\mathbf{x}$ is drawn (Corduneanu et Jaakkola, 2006). In particular, if $\mathbf{x}$ lies in a metric space, then limits can be taken, leading to an approximation of the limiting case where regions, defined as hypercubes, are both vanishingly close and small. The form of the regularizer in the parametric case using empirical estimates of $p(y|\mathbf{x})$ closely matches minimizing the Fischer information of the resulting conditional over $\mathbf{x}$. This has the effect of preventing $p(y|\mathbf{x})$ from varying too much with model parameters where the probablity of $\mathbf{x}$ is high, that is, the model should not tell much more about the label than the region already imparts. In regions of low density, however, the model is allowed to be more informative.

Graph based methods are related to the information regularization of Corduneanu and Jakkola in that in the diagnostic setting, discriminant functions are regularized by smoothness penalties defined by regions, more precisely defined as possibly weighted graph neighborhoods. In the transductive setting, a simple method presented by Zhu and Ghahramani called label propagation (Zhu et Ghahramani, 2002), is an iterative algorithm in which at each iteration, an unlabeled vertex is labeled with the weighted average of each of its neighbors. Bengio *et al.*present a modification in which each unlabeled point is again labeled with a weighted average of its neighbors but with a small regularization term which can be thought of as a smoothing prior and labeled points are allowed to change as well, but the original label is also included in the average (Bengio *et al.*, 2006). Zhu and Gramini also present a modification called label spreading in which the labeled data is also allowed to change. Again, each label is allowed to change as a weighted sum with the original label as a component. Depending on a constant parameter, the original label may take more or less weight. Instead of using the Laplacian, they also use the normalized Laplacian, in which the degree is normalized to 1. As such, the normalized Laplacian of an undirected graph also can be interpreted as a scaled stochastic matrix with unit eigenvalues. Perhaps because of this, Szummer and Jaakkola present a method in which each example is treated as a vertice in a transition matrix, with unlabeled data labeled by the probability of hitting that vertice at time t starting from labeled vertices in a Markov random walk. This is somewhat unintuitive, since both t and the initial probability vector are both somewhat arbitrary, as Bengio *et al.*comment (Bengio *et al.*, 2006).

By inspection, Bengio and Dellalau show that the natural extension of label propagation to graph regularization is given by minimizing the weighted difference between neighboring

vertices (Bengio *et al.*, 2006). When the difference is squared this results in a quadratic cost which can be expressed succinctly in matrix form in the diagonal label matrix and the Laplacian. One way to enforce this penalty is to view the graph Laplacian as an approximation to the Laplacian on a smooth manifold in Hilbert space, whose eigenfunctions provide a basis for all square differentiable functions on the manifold. Projecting functions onto the space of the eigenfunctions of the manifold Laplacian provides a way of smoothing functions. Since regularization seeks smooth functions, Sindhwani *et al.*suggest using the eigenvectors corresponding to the smallest eigenvalues of the graph Laplacian as a basis for the imputed labels (Sindhwani *et al.*, 2006). The imputed label is viewed as a function which consists of a linear combination of these eigenvectors (which are the smoothest based on the eigenvalues).

An alternative is to apply the results in the dual space, leading to the Laplacian regularized least squares (LapRLS) from Sindhwami et al (Sindhwani *et al.*, 2006). This is actually quite similar to the quadratic cost term proposed by Bengio *et al.*, albeit posed in terms of regularization theory.

The RLS problem can be posed as a choosing a function which minizes a cost, typically squared loss, subject to some smoothness constraints. For any Mercer kernel $\mathbf{K}$, there exists an associated RHKS, H_k and norm. By the Representer theorem, the solution to the RLS objective admits a finite representation as a linear combination of kernel functions. The natural extension of unlabeled data is to penalize the smoothness of the function in both the marginal distribution $P(\mathbf{x})$, as well as the ambient space, H_k. Sindhwami argue that the natural penalty on the manifold is the Laplace operator, and that this can be approximated by the graph Laplacian. The effect is to add an additional term which penalizes the smoothness of the function on the adjency graph. Comparing this to the quadratic cost criteria, we see that the two are identical where the graph penalty norm is the Laplacian. A similar design is also discussed for the SVM, which involves adding this penalty to the SVM loss objective which also admits a finite representation. Sindhawami *et al.*also show that the graph smoothness penalty can also be combined with a kernel function to form a new RHKS associated with a norm, in effect creating a unlabeled-data dependent kernel.

The above is generalized in (Zhou et Scholkopf, 2006), who argue that different parameterizations of discrete graph gradients determine various graph regularization penalties. They introduce a p-Dirichelet form of a graph gradient norm, and show that when $p = 2$, it results in a normalized Laplacian operator. Using a slightly different form of the graph gradient results in standard graph Laplacian. The more general discrete graph gradients presented by Zhou and Scholkopf represent a rigorous view of graph regularization, allowing for regularizations of higher order derivatives as well as insight into the difference between regularizations.

One issue so far has been that the graphs discussed above have been assumed to be undirected. One way of dealing with directed edges is in the powerful framework of graphical models. In this case each example remains a vertex with directed edges modeling dependence relationships. In the Conditional Harmonic Mixing model (CHM) of Burges and Platt, the label at node i is equivalent to a discrete random variable associated with conditional probability table for each of its edges. At each iteration, CHM, minimizes the KL divergence between a current posterior distribution and that induced by its neighborhood. Given a particular graph structure, the result can be expressed as a linear system, which provides a method of determining conditional probabilities given initial posterior probabilities which Burges and Platt refer to as "CHM learned" (Burges et Platt, 2006). For a document classification task, the documents are expressed as bags of words and directed links are defined by the k-nearest neighbor rule. That is, a directed edge between i and unlabeled j exist if i is one of the k nearest neighbors of j for different values of $k \leq K$. Conditional probability tables are shared for all links $k \leq K$. Note that using unit conditional probabilities amounts to a directed, probabilistic version of label propagation. Burges and Platt use an SVM on labeled points to determine initial posteriors and show increased performance when using as little as 10 training examples per class, but decreased performance when using a larger proportion. The authors argue that graph construction is quite important for improved performance, as the learned CPT's do not appear to be much more informative than the unit CPT baseline.

2.3.6 Weakly Labeled Learning

Up to now, all of these models are presented under the assumption that the function $y = f(\mathbf{x})$ is estimated where $\mathbf{x}$ and y describe the data completely. However, stronger assumptions may also appear. One additional source of information is the label proportion. That is, labels are not usually distributed uniformly in the unlabeled set. Along with TSVM's, in which the learned classfier can be constrained to label the unlabeled data in a particular proportion, graph based methods may also make use of label proportions. For label propagation and label spreading, this can be accomplished simply by adding a bias term to the weighted means. For CHM and Laplacian regularization, additional labeled nodes set to the label proportion can be added accompanied by appropriately weighted edges to unlabeled nodes. The strength of the weights, or the weight of the bias term, are determined by by deciding how much the class proportion should matter. In the TSVM case, such proportions can be set as a hard constraint.

The label proportion gives rise to the idea of weak labels, since a high proportion could be viewed as a weak label. A related approach is presented by Mann *et al.*called the Generalized EM Criteria (GEM) (McCallum *et al.*, 2007). In the GEM, the unlabeled data and label

is also associated with a relationship $z(y, x)$. In the GEM model, z is a function in which dependence is not strictly defined through a directed edge. The function z is observed, and thus the posterior over any model parameters $p(\theta|D)$ should be constrained to respect the expectation $\frac{1}{U}\sum_u^U z(\mathbf{y}_u, \mathbf{x}_u) = \int_x \sum_y z(x, y)p(y|x)p(x) = \mathbb{E}[z]$. In essence, the model behavior is similar to moment matching.

For each example $z(\mathbf{y}_u, \mathbf{x}_u)$ is not required to be known, only an empirical average. Mann et McCallum (2007), gives an algorithm where $z(y, x)$ depends only on y and an element of the $\mathbf{x}$ vector, x_k. They call this a "labeled feature." Furthermore, they define $z(y, x) = p(y|x_k = 1)$. The empirical label proportion conditioned on $x_k = 1$ must be obtained from outside the model, as in an sample estimate.

To enforce this constraint, Mann *et al.*. use a KL divergence term between the estimate and the model predictions. That is, letting $\hat{p}$ denote the estimate (a vector of class proportions), the GEM is formulated as $KL(\hat{p}||q)$, where q is a distribution induced by the model $q(y) = \sum_{\mathbf{x}_u:\mathbf{x}_{uk}=1} p_\theta(y|\mathbf{x}_u)$. This term is added as a regularizer to the standard penalized multinomial regression model. Note that the distribution q is multinomial and the distribution that minimizes the KL term is $\hat{p}$. The KL divergence is then $\sum_y \hat{p} \log \frac{\hat{p}}{q(y)}$. This can be decomposed into an entropy formulation where the final regularizer, ignoring constant terms, is given as the cross entropy

$$-\sum_y \hat{p}(y) \log q(y) \tag{2.25}$$

The authors also use a temperature T in $p(y|x) \propto \exp(\frac{1}{T})$, with $T < 1$, to attribute more mass to particular classes by weighting the logistic function in order to avoid degenerate solutions in which the vector of outputs is $\hat{p}(z)$ for every class, while keeping the objective differentiable. The above algorithm is called label regularization, as if focuses on prior beliefs about the relationship of labels to specific features. One way to interpret this regularization is as a test. The learned distribution is tested against a known relationship, and penalized for divergence.

The regularization, however, need not be defined as a KL divergence measure. The GEM is related to the co-training framework of Blum and Mitchell in that $p(y|x_k)$ can be thought of as the result of an alternative classifier trained on a separate view of the data (Blum et Mitchell, 1998). However, clearly $\mathbf{x}$ and x_k are not necessarily conditionally independent given the labels, as required in co-training. On the other hand, by using an out-of-sample estimate or prior knowledge, the information of the final classifier does not depend on a particular view of the data, making $p(y|x_k)$ similar to an alternative classifier, albeit an extremely weak classifier except in certain cases.

CHAPTER 3

Sparse Kernel Methods

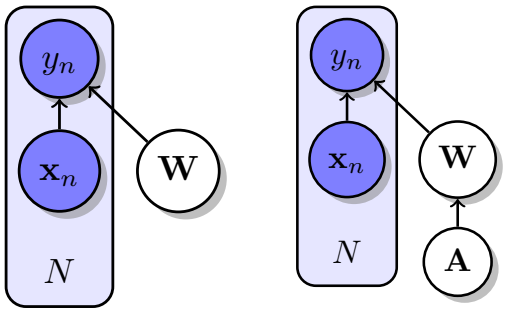


Figure 3.1 Graphical model of supervised learning. $\mathbf{x}$ is the input, y is the label, $\mathbf{W}$ denotes model parameters. 3.1(a) depicts standard setting, in 3.1(b), $\mathbf{A}$ represents an additional hyper-parameter which is also equipped with a distribution.

This chapter introduces our model for sparse kernel learning. The following chapters describe how it can be adapted to additional sources of information such as weak or noisy labels. The supervised model discussed in this section, however, serves as the basis from which a model for weakly-supervised learning is derived.

A fully supervised model can be illustrated in a graphical model as shown in Figure 3.1(a). This basic model describes a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, where $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$, and $y = \{y_n\}_{n=1}^N$. Here, y_n refers to labels and $\mathbf{x}_n$ an input or feature vector. The model parameters are $\mathbf{w}$, which in the Bayesian setting, are treated as hidden variables. The plate notation indicates conditional independence, where there are N pairing of variables, so that

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_n^N p(y_n|\mathbf{x}_n, \mathbf{w}), \tag{3.1}$$

In the strictly supervised setting, specifying a conditional distribution and then maximizing the log-likelihood with respect to $\mathbf{W}$ is a common approach, also called maximum likelihood (ML). A common binary probabilistic discriminative model is logistic regression, in which the optimal $\mathbf{w}$ is set to the arg max of the log probability

$$\mathbf{w} = \text{argmax}_{\mathbf{w}} \log \prod_{n}^{N} p(y_n|\mathbf{x}_n, \mathbf{w}) = \text{argmax}_{\mathbf{w}} \sum_{n}^{N} \log \frac{1}{1 + \exp(-y_n \mathbf{w}^T \phi(\mathbf{x}_n))}. \tag{3.2}$$

Here, ϕ refers to a transformation or feature function of $\mathbf{x}$, in the case where the input is subject to a transformation first. This includes use of kernel functions, in which $\phi(\mathbf{x}_i) = (K(\mathbf{x}, x_i))_{i=1}^{N}$, for some kernel function K. For example, in the linear kernel, ϕ could be the cross product $\mathbf{x}_i^T \mathbf{x}$. In this case, the logistic regression is also known as kernel logistic regression, (Jaakkola et Haussler, 1999), and in a full kernel matrix, the dimensionality of $\phi(\mathbf{x}_n)$ is N, meaning that $\mathbf{w}$ has N elements.

Moreover, the prior can also be augmented with a hyperprior, a distribution for the vector α, which can lead to useful properties. As shown in Figure 3.1(b), the inclusion of this variable yields the joint distribution

$$p(\mathbf{y}, \alpha, \mathbf{w}|\mathbf{X}) = \prod_{n}^{N} p(y_n|\phi(\mathbf{x}_n), \mathbf{w}) p(\mathbf{w}|\alpha) p(\alpha) \tag{3.3}$$

In the Relevance Vector Machine (RVM), Tipping *et al.*showed that a Gamma hyperprior leads to sparse weights $\mathbf{W}$ (Tipping, 2000a). The combination of a Gaussian prior with a hyperprior on the covariance results in alternative prior formulations, after integrating out the hyper-parameter.

3.1 Sparse Priors

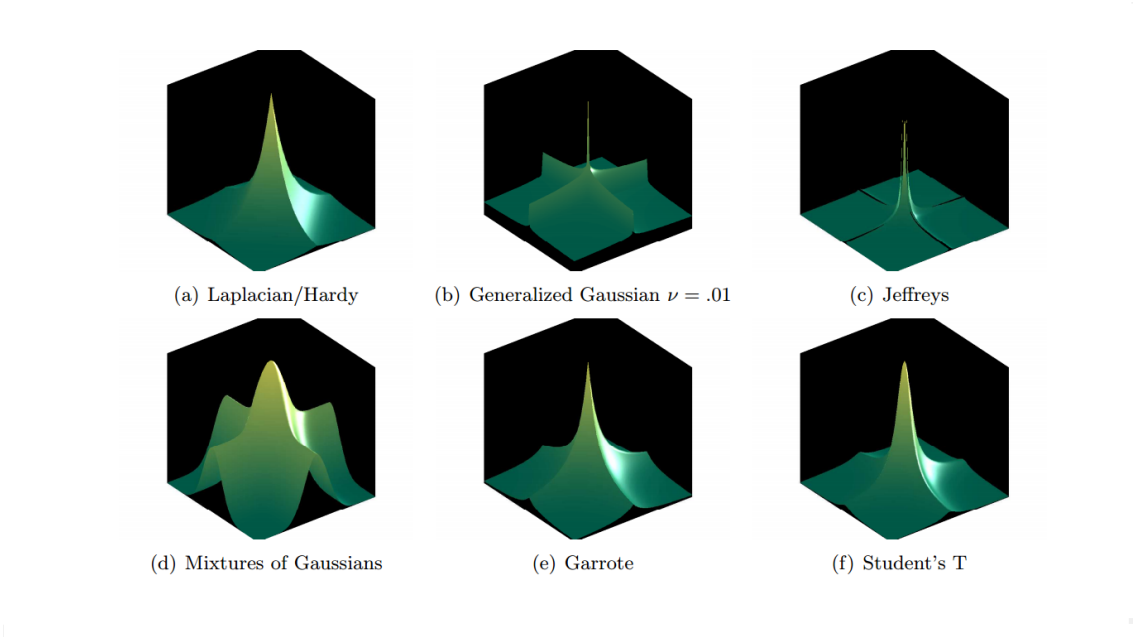


Figure 3.2 Visualization of some sparse prior distributions, details in text

Table 3.1 Listing of sparse priors. For a discussion of Gaussian Mixtures and the Garrote, see Appendix A.

Laplace	$\propto \exp(-\gamma\lvert\mathbf{w}_n\rvert),\ \gamma > 0$
Hardy	$\propto \exp(-\gamma(\sqrt{\beta^2 + \mathbf{w}_n^2})),\ \gamma \to 0,\ \beta > 0$
Jeffreys	$\propto \frac{1}{\lvert\mathbf{w}_n\rvert}$
Exponential	$\propto \exp(-\frac{1}{2\alpha_n} w_n^2) \exp(-\gamma\lvert\alpha_n\rvert),\ \gamma > 0$
Generalized Gaussian	$\propto \exp(-\frac{1}{2\alpha_n} w_n^2) \exp(-\gamma\lvert\alpha_n\rvert^\nu),\ 0 < \nu < 2,\ \gamma > 0$
Gaussian Mixture	$\propto \sum_k^n \pi_k \mathcal{N}(\mathbf{w}_n\lvert 0, \gamma_k^2),\ \pi_k > 0,\ \sum \pi_k = 1$
Garrote	$\propto \exp\left(\frac{\mathbf{w}_n^2}{4\sigma^2} - \frac{\lvert\mathbf{w}_n\rvert\sqrt{\mathbf{w}_n^2+4\gamma^2}}{4\sigma^2} - \frac{\gamma^2}{\sigma^2}\text{arcsinh}\left(\frac{\alpha_n}{2\gamma}\right)\right),\ \gamma > 0$
Student's t	$\propto \int N(\alpha_n\lvert 0, \gamma^{-1}) \prod \Gamma(\gamma, a, b) d\gamma,\ a \geq 0,\ b \geq 0$

In addition to the Gamma hyperprior, there are a number of distributions which typically lead to sparsity. In fact, the hyperprior can be interpreted as a parameter, as this choice leads to different sparse solutions. Table 3.1 lists a few well-known sparsity-inducing priors.

Visualization of each sparse prior is given in figure (3.2). In viewing the sparse priors as a group, the commonality is heavy tails – each is "peaked" at zero with a decay in probability that is sub-exponential.

In the following sub-sections, algorithms using the Laplacian, Exponential, Jeffrey's, Generalized Gaussian are developed and discussed. Appendix A describes experiments using a Mixtures of Gaussians Prior and a procedure based on the Non-Negative Garrote. All experiments are done with respect to a simulated mixture distribution described in (Hastie *et al.*, 2005, p. 12). The dataset consists of 100 examples drawn from two mixtures of Gaussians. All validation is done on a held-out set generated from the means. Testing and training is done on a 50-50 split of the original data.

3.1.1 Laplacian

The use of the Laplacian and Hardy priors essentially generalize l_1 regularization, (with the Generalized Gaussian also equivalent under these models as $\nu = 1$), as shown in Table (3.1). Maximum a priori (MAP) estimation – maximizing the posterior probability – under this class of prior has been seen under various guises in the context of regularized kernel linear regression usually called the LASSO, (Hastie *et al.*, 2005). The form of the logistic optimization problem using MAP estimation becomes

$$\mathbf{w} = \text{argmax}_{\mathbf{w}} \sum_{n}^{N} \log \frac{1}{1 + \exp(-y_n \mathbf{w}^T \phi(\mathbf{x}_n))} + \lambda \sum_{n}^{N} |w_n|, \tag{3.4}$$

which is also known as l_1 regularization.

The Hardy prior functionally differs from the Laplacian in that a perturbation term is included, β, *c.f.*Table (3.1), to offset the discontinuity at zero. The l_1 relaxation leads to a convex objective, as $f(\mathbf{y}, \mathbf{w})$ is taken to be sum of squared errors and as such is convex. Secondly w_i can be optimized independently of other w. This results in an iterative scheme where an individual weight w_i is removed from the model (or "shrunk" in LASSO terminology) based on the criteria of its effect on the model fit term and the sparsity term.

Letting $f(\mathbf{y}, \mathbf{w})$ be the original function to optimize, *i.e.*Equation (3.4), using a 1st order Taylor expansion around $\mathbf{w}^{(t)}$, (the estimate of $\mathbf{w}^*$ at time t), the optimization function f is approximated with

$$f(\mathbf{y}, \mathbf{w}) = f(\mathbf{y}, \mathbf{w}^{(t)}) - (w - w^{(t)})^T \nabla f(\mathbf{y}, \mathbf{w}^{(t)}). \tag{3.5}$$

In terms of $\mathbf{w}^{(t)}$, the update equation is

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} f(\mathbf{y}, \mathbf{w}^{(t)}) + (\mathbf{w} - \mathbf{w}^{(t)})^T \nabla f(\mathbf{y}, \mathbf{w}^{(t)}) + \lambda \sum |w_i|. \tag{3.6}$$

However, this expansion is only accurate in neighborhoods around $\mathbf{w}^{(t)}$, so a penalty term is included, $\|\mathbf{w} - w_j\|$

$$\begin{aligned}\mathbf{w}^{(t+1)} &= \text{argmin}_{\mathbf{w}} f(\mathbf{y}, \mathbf{w}^{(t)}) \\ &+ (\mathbf{w} - \mathbf{w}^{(t)})^T \nabla f(\mathbf{y}, \mathbf{w}^{(t)}) + (1/2\delta^2)(\mathbf{w} - \mathbf{w}^{(t)})^T(\mathbf{w} - \mathbf{w}^{(t)}) + \lambda \sum |w_i| \end{aligned} \tag{3.7}$$

Noting that the above contains many terms that are constant, a more convenient representation is

$$\mathbf{w}^{(t+1)} = \text{argmin}_{\mathbf{w}} \lambda \sum |w_i| + \|\mathbf{w} - (\mathbf{w}^{(t)} - \delta^{(t)} \nabla f(\mathbf{y}, \mathbf{w}^{(t)}))\|^2 \tag{3.8}$$

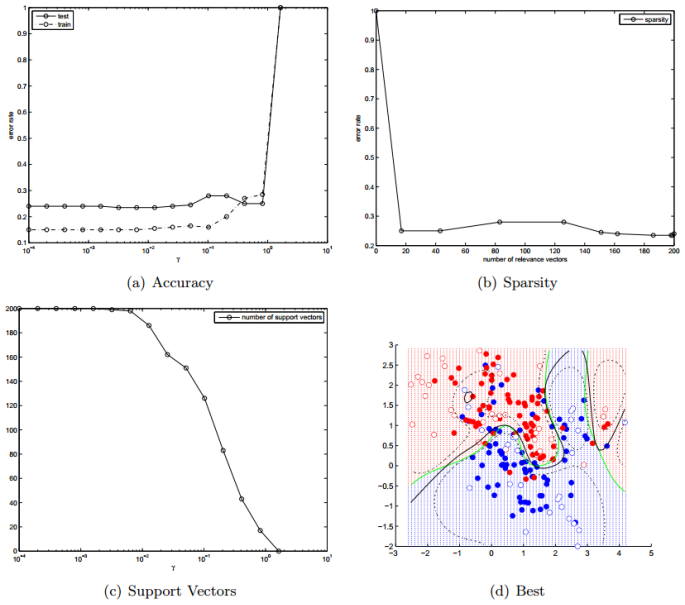
Thus δ here specifies a step size, and can be specified or set using line searches. The basic iterative approach is then a subproblem at time t. The subproblem itself can be broken into subproblems in individual w_j (provided f is sufficiently separable). That is, taking derivatives and setting to 0, the update for an individual weight is

$$w_j^{(t+1)} = (\mathbf{w}^{(t)} - \delta^{(t)} \nabla f(\mathbf{y}, \mathbf{w}^{(t)}))_j \pm \lambda. \tag{3.9}$$

This leads to the shrinkage operation usually used in sub-gradient methods. Letting $h_j = (\mathbf{w}^{(t)} - \delta^{(t)} \nabla f(\mathbf{y}, w^{(t)}))_j$,

$$w_j^{(t)} = \begin{cases} h_j - \delta^{(t)}\lambda & h_j \in (\delta^{(t)}\lambda, \infty) \\ 0 & h_j \in [-\delta^{(t)}\lambda, \delta^{(t)}\lambda] \\ h_j + \delta^{(t)}\lambda & h_j \in (-\infty, -\delta^{(t)}\lambda) \end{cases} \tag{3.10}$$

However, in practice, minimization of the l_1 norm under the Laplacian prior in KLR does not generally yield sparse solutions. An analysis of this is shown in figure (3.2) and figures (3.3).



(a) Accuracy (b) Sparsity

(c) Support Vectors (d) Best

Figure 3.3 Laplace prior : accuracy, sparsity and number of support vectors (NSV).

Table 3.2 Best Results for MAP estimate with Laplace prior. NSV is the number of support vectors or non-zero weights.

	training error	testing error	Bayes error	NSV	γ
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
Laplace	.150	.235	.19	198	.0064

3.1.2 Exponential

On the other hand, an alternative objective under the Exponential hyper-prior is possible if we decompose the prior. That is, if w_j is distributed as $N(w_j|0, \tau_j)$ and allow each τ_j to have independent exponential distributions, parameterized by λ.

$$p(\tau_j|\lambda) = \frac{\lambda}{2} \exp\big(-\frac{\lambda\tau_j}{2}\big) \tag{3.11}$$

Under kernel logistic regression, data $\mathbf{X}$, is represented as the vector of kernel function evaluations $\phi(\mathbf{u}, \vee)$. These features can be represented as a matrix $\mathbf{K}$, where $\mathbf{K}_{ij} = \phi(\mathbf{x}_i, \mathbf{x}_j)$, called the kernel matrix. Under the kernel logistic regression objective the complete log likelihood is now, with $\mathbf{A} = diag(\tau^{-1})$),

$$\log p(\mathbf{y}, \mathbf{w}, \tau) \propto \sum \log \sigma(y_i \mathbf{K}_i \mathbf{w}) - \frac{1}{2}\mathbf{w}^T \mathbf{A} \mathbf{w} - \frac{\lambda}{2} \sum |\tau_j|. \tag{3.12}$$

Here $\mathbf{K}_i$ is the *row* vector in the symmetric $\mathbf{K}$ matrix.

To maximize this, the standard EM algorithm treats $\mathbf{w}$ as a parameter, and maximizes $\mathbf{w}$ under the expectation of $p(\tau|\mathbf{y}, \mathbf{w}^{(t)})$.

$$\mathbf{w}^{(t+1)} = \text{argmax}_{\mathbf{w}} \mathbb{E}_{p(\alpha)}[\log p(\mathbf{y}, \tau|\mathbf{w})] = \sum \log \sigma(y_i \mathbf{K}_i \mathbf{w}) - \frac{1}{2}\mathbf{w}^T \langle \mathbf{A}|\mathbf{w}^{(t)}\rangle \mathbf{w} \tag{3.13}$$

As the above maximization with respect to $\mathbf{w}$ does not depend on the last term on the right hand side of (3.12), we have only to maximize with respect to this expectation. Again, k denotes the iteration index. Furthermore, $\langle \tau_j|\mathbf{w}^{(t)}\rangle = \int \tau_j p(\tau_j|y, \mathbf{w}^t) d\tau_j$. Since τ_j only depends on $w_j^{(t)}$, we have $\tau_j = \int p(\tau_j|w_j^{(t)})\tau_j d\tau_j$. Before getting to the conditional expectation of $\mathbb{E}_{\mathbf{w}}[\tau_j] = \int \tau_j p(\tau_j|w_j^{(t)}) d\tau_j$, let us inspect the form of the prior after integrating out τ_j. This is $p(w_j)$. The integral in question will be necessary to derive the conditional expectation, as will be seen shortly.

$$p(w_j|\tau_j) = \frac{\lambda}{2\sqrt{2\pi}} \int_0^\infty \tau_j^{-1/2} \exp\big(-\frac{w_j^2}{2\tau_j}\big) \exp\big(-\frac{\lambda\tau_j}{2}\big) d\tau_j. \tag{3.14}$$

Letting $\alpha_j = \frac{1}{\tau_j}$, where $d\tau_j = -\alpha_j^{-2} d\alpha_j$,

$$p(w_j) = \frac{\lambda}{2\sqrt{2\pi}} \int_0^\infty \alpha_j^{-\frac{3}{2}} \exp\big(-\frac{\alpha_j w_j^2}{2}\big) \exp\big(-\frac{\lambda}{2\alpha_j}\big) d\alpha_j. \tag{3.15}$$

The above integral is the Laplace transform of a Levy (stable distribution) with exponent $\frac{1}{2}$, and so the required probability is given by Cartea among others (Feller, 1971), (Cartea et Howison, 2003) as

$$p(w_j) = \frac{\lambda}{2\sqrt{2\pi}}\mathbb{E}_f[\exp(-\mathbf{s}\alpha_j)] \tag{3.16}$$

$$s = \frac{w_j^2}{2} \tag{3.17}$$

$$f = \alpha_j^{-\frac{3}{2}} \exp\big(-\frac{\lambda}{2\alpha_j}\big). \tag{3.18}$$

$$\tag{3.19}$$

Therefore,

$$\mathbb{E}[\exp(-\mathbf{s}\alpha_j)] = \sqrt{\frac{2\pi}{\lambda}} \exp\big(-\frac{\sqrt{\lambda w_j^2}}{2}\big) \tag{3.20}$$

$$p(w_j) = \frac{\sqrt{\lambda}}{2} \exp\big(-\frac{\sqrt{\lambda}}{2}|w_j|\big) \tag{3.21}$$

Now noting that the integral is a Laplace transform, and that normalization of the conditional probability $p(\alpha_j|w_j)$ is given by the Laplace transform above, we can use the following identity :

$$\int_0^\infty \exp(-st)t^n f(t)dt = (-1)^n \frac{\partial}{\partial s}\int_0^\infty \exp(-st)f(t)dt \tag{3.22}$$

$$\tag{3.23}$$

With the required transform being

$$\mathbb{E}_{p(\alpha_j)}[\exp(-\mathbf{s}\alpha_j)] = \frac{\sqrt{2\pi}}{2|w_j|} \exp\big(-\frac{\sqrt{\lambda}}{2}|w_j|\big) \tag{3.24}$$

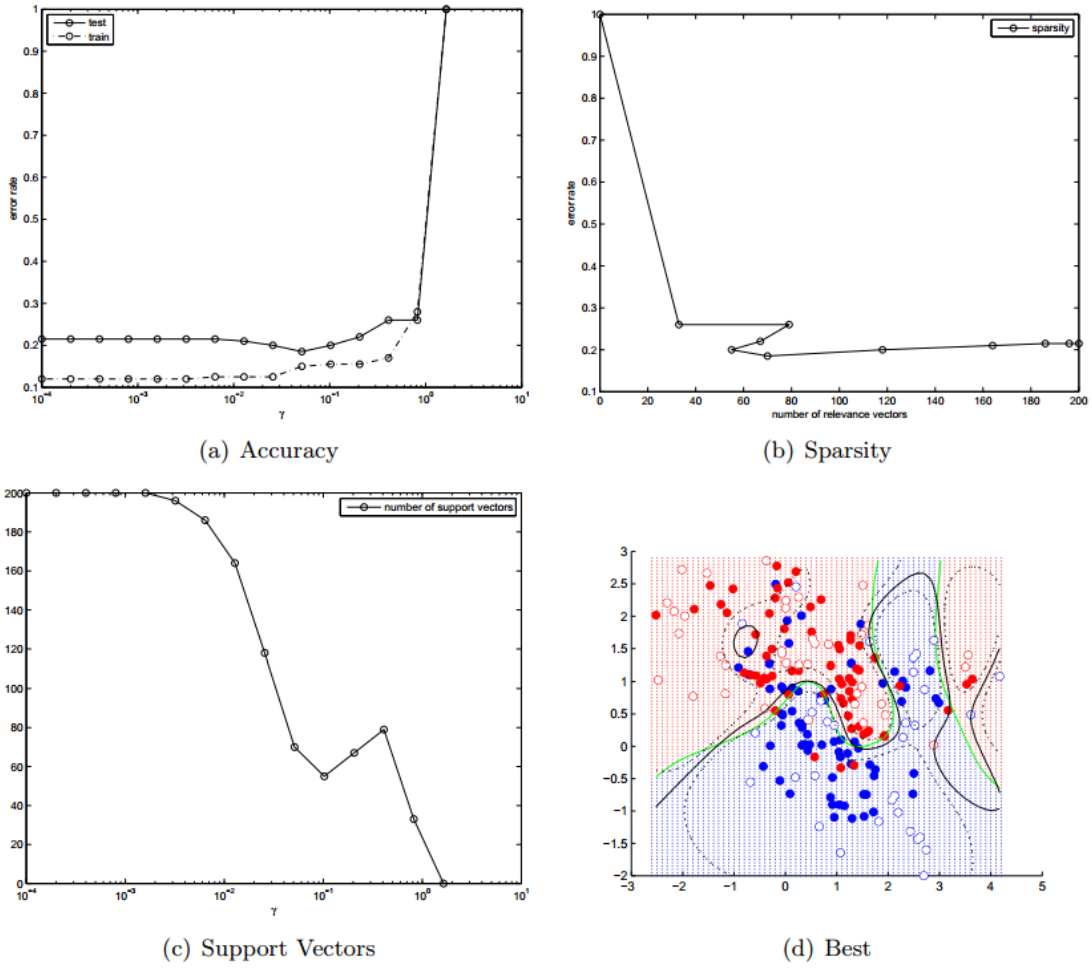
So that after multiplying by $\frac{\lambda}{2\sqrt{2\pi}p(w_j)}$,

$$\langle \alpha_j | w_j^{(t)} \rangle = \frac{\sqrt{\lambda}}{2|w_j|} = \frac{\gamma}{|w_j|} \tag{3.25}$$

Here $\alpha_j = \tau_j^{-1}$. To recapitulate, first we randomly assign α_j, then maximize $\mathbf{w}$ by iteratively re-weighted least squares or some other means and take the expectation of α_j by

(3.25).

Results using this model for varying λ are shown in 3.4, with the best results compared against SVM in 3.3. Since we are not integrating over the full posterior, this cannot be called a Bayesian procedure, rather it is a form of Variational Bayes which is a form of approximate Bayesian inference (Beal, 2003).



(a) Accuracy (b) Sparsity

(c) Support Vectors (d) Best

Figure 3.4 Exponential prior : accuracy, sparsity and number of support vectors

Table 3.3 Exponential prior : Best Results

	training error	testing error	Bayes error	NSV	γ
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
Exponential	.155	.200	.19	55	.1024

3.1.3 Jeffreys

One issue with the above is the additional parameter γ. It is possible to avoid cross-validation with alternate priors, or if a so-called uninformative prior is used. One such uninformative prior is Jeffreys' prior, which potentially induces more sparsity. In viewing figure (3.2), we see that it is similar to the Laplacian except that the density at 0 is not finite.

(Figueiredo *et al.*, 2002) notes that the Jeffrey's prior is proportional to the square root of Fisher information score – when α_i is 0, no information about w_j, the weight vector exists, because the integral is unbounded (Figueiredo *et al.*, 2002). Letting $\tau_j = \alpha_j^{-1}$, again,

$$p(\tau_j) = \sqrt{\frac{2}{\tau_j^2}} \tag{3.26}$$

The algorithm is again the same, except that

$$\mathbb{E}[\alpha_j | \mathbf{w}^{(t)}] = \frac{1}{|w_j|^2} \tag{3.27}$$

The math is a bit simpler here, although based on the same ideas as the Exponential, so it will be omitted here. Results are also shown in figure (3.5) and figure (3.4)

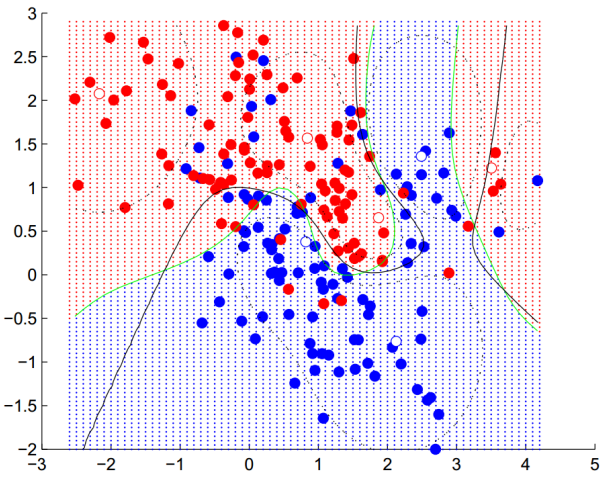


Figure 3.5 MAP classifier with Jeffrey's prior

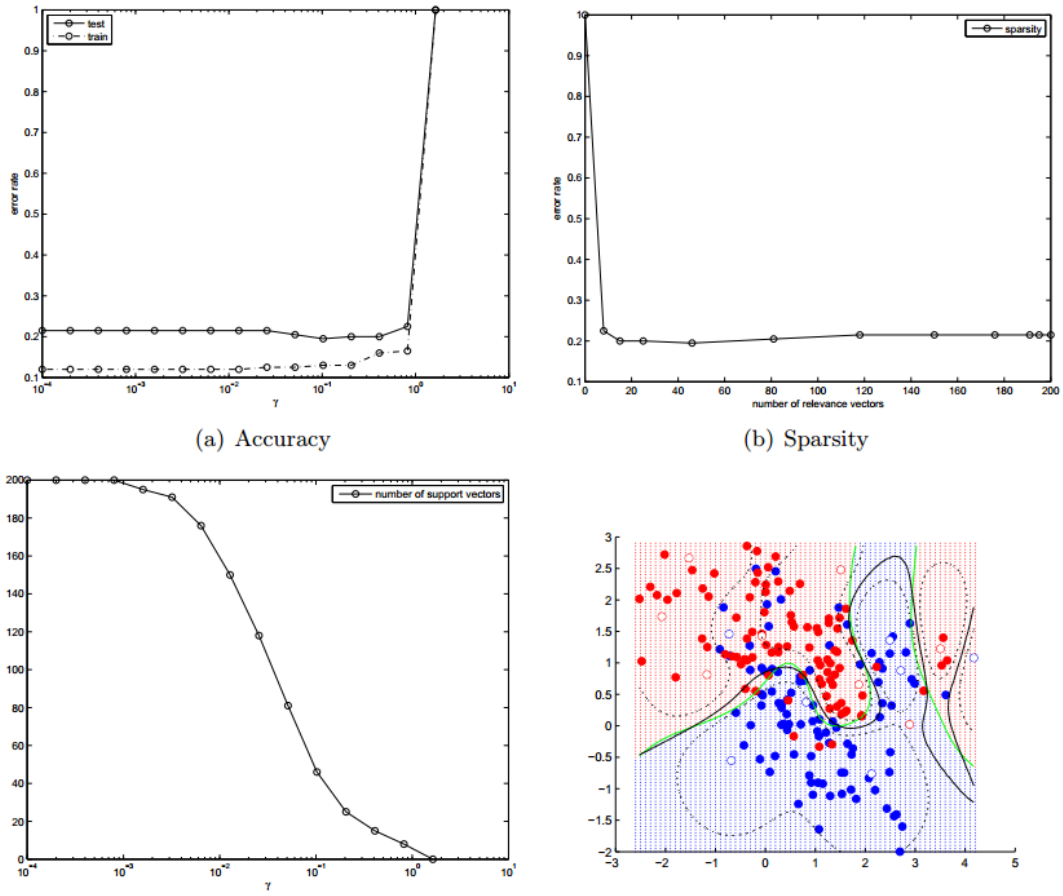
Table 3.4 Jeffrey's prior results

	training error	testing error	Bayes error	NSV	γ
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
Jeffrey's	.155	.225	.19	7	-

3.1.4 Generalized Gaussian

(Wipf, 2006) showed that the update equations derived for the Exponential and Jeffreys priors can be generalized by adding a parameter p, with $\mathbb{E}[\alpha_j|\mathbf{w}, \lambda] = \frac{\lambda}{|w_j|^{(2-p)}}$. This gives rise to the generalized Gaussian, where changing λ and p (with $p \in (0, 1]$), gives a smooth parameterization between the l_0 and l_1 norms. With $\lambda = 1$, $p \to 0$, the result approaches the l_0 norm, and can therefore interpolate to choose between convexity profiles. David Wipf showed that as $p \to 0$, many local maxima exist, but are guaranteed to provide a maximally sparse solution and conversely, as $p \to 1$, we have fewer local maxima, but no guarantee of convergence to a maximally sparse solution (Wipf, 2006).

(Wipf, 2006) recommend using this trade-off to obtain solutions according to a desired convexity and sparsity. However, their results are asymptotic, which make utilizing these facts difficult in practice. In light of the above two results, the Jeffrey's prior, in which $p = 0$, might be suffering from poor model fit. In experiments, setting $p = .1$ provided best results on this data. The results for varying γ with $p = .1$ are shown in figures (3.6) and (3.5).



(a) Accuracy (b) Sparsity

(c) Support Vectors (d) Best

Figure 3.6 Generalized Gaussian

Table 3.5 Best Resulting MAP estimate with Generalized Gaussian

	training error	testing error	Bayes error	NSV	γ
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
Generalized Gaussian	.160	.200	.19	15	.4096

3.1.5 Sparse Bayesian Learning

In the preceding, we were learning MAP estimates of the posterior. Sparse Bayesian Learning (SBL), as defined by (Wipf, 2006), is based on evidence approximation where we maximize the probability of the data, and treat $\mathbf{w}$ as a variable and integrate it out. SBL is typified with the use of a Gaussian prior and a non-informative or conjugate hyperprior. That is, the optimization is of the following,

$$p(\mathbf{y}|\alpha) = \int_{\mathbf{w}} p(\mathbf{y}|\mathbf{w})N(\mathbf{w}|\mathbf{0}, \alpha^{-1})p(\alpha)d\alpha. \tag{3.28}$$

The implicit prior in the original RVM is uninformative, and the hyperprior is implicitly a Jeffrey's distribution. This is the original formulation of the RVM. It is important to note that this is an *ML* estimate of $p(\mathbf{y}|\alpha)$. Meanwhile the predictive distribution is given by $p(\mathbf{y}|\langle\mathbf{w}\rangle)$, rather than the $\int_{\mathbf{w}}\int_{\alpha} p(\mathbf{y}|\mathbf{w}, \alpha)d\mathbf{w}d\alpha$ that a Bayesian might insist on. However, because this thesis is concerned with building sparse, and therefore fast and small classifiers, the full integration is somewhat at odds with the purpose of the final classifier.

Maximizing the evidence, however, is here intractable so $p(\mathbf{y}|\alpha)$ is approximated using the Laplace approximation with respect to $\mathbf{w}$. Comparing this with the discussion of the Jeffrey's prior above, the difference is in the choice of optimization. Again, rather than optimizing the posterior $p(\mathbf{w}|\mathbf{y}, \alpha)$ SBL optimizes $p(\mathbf{y}|\alpha)$.

As the above uses point estimates, it is closely related to a MAP estimate of the evidence using a Jeffrey's hyperprior. Variational Bayes is an alternative to these point estimates, approximating the posterior using a factorization of the posterior distribution. As I have already written about the RVM as well as the fast RVM, I will not present a discussion here.

3.1.6 Variational Bayes

In contrast to the RVM using point estimates, the Variational RVM (VRVM), as presented by Bishop (Bishop et Tipping, 2000), includes a hyperprior which allows for a factorisable approximating posterior. This procedure is also known as Variational Bayes (Jaakkola et Jordan, 1997), (Jordan *et al.*, 1999b), (Beal, 2003). The basic idea in Variational Bayes is to approximate the posterior using a lower bound.

In the VRVM formulation, the hyperprior on the precision is a Gamma hyperprior, (the resulting prior $p(w)$ is Student-T). As such the joint probability is given by

$$p(D, \theta) = \prod^{n} p(y_i|\mathbf{K}_i\mathbf{w}) \prod^{n} p(w_j|\alpha_j)p(\alpha_j|a, b) \tag{3.29}$$

$$= \prod \sigma(y_i\mathbf{K}_i\mathbf{w})N(w_j|0, \alpha_j^{-1})\Gamma(\alpha_j|a, b) \tag{3.30}$$

EM will attempt to minimize the KL divergence between an approximation measure q and the true measure p, so that the error function is given by the equality that holds for any distribution q – $\log p(\mathcal{D}) = L(q) + KL(q||p)$, where $L(q)$ is defined as

$$L(q) = \int q(\theta) \log \frac{p(\mathcal{D}, \theta)}{q(\theta|\mathcal{D})}, \tag{3.31}$$

for a measure q over variables $\mathcal{D}, \theta$, and an alternative measure p over these same variables, and KL is the Kullback-Liebler divergence

$$KL(q||p) = \int q(\mathcal{D}, \theta) \log \frac{q(\mathcal{D}, \theta)}{p(D|\theta)}, \tag{3.32}$$

Since $\log p(\mathcal{D})$ is independent of q, maximizing $L(q)$ is equivalent to minimizing $KL(q||p)$. Since the KL divergence ≥ 0, this provides a lower bound on $\log p(\mathcal{D})$. In the case of the RVM, $\theta = (\mathbf{w}, \alpha)$, and so the error function is

$$\mathcal{E} = -\int_{\mathbf{w}} \int_{\alpha} q(\mathbf{w}) q(\alpha) \log \frac{\prod \sigma(y_i \mathbf{K}_i \mathbf{w}) \prod N(w_j, 0, \alpha_j^{-1}) \Gamma(\alpha_j | a, b)}{q(\mathbf{w}) q(\alpha)} \tag{3.33}$$

To minimize $\mathcal{E}$ with respect to $q(w)$, we note that $\mathcal{E}$ as a functional of $q(\mathbf{w})$ is

$$\int_{\mathbf{w}} q(\mathbf{w}) \big[-\sum^{n} \log \sigma(y_i \mathbf{K}_i \mathbf{w}) + \sum^{k} \frac{\langle \alpha_j \rangle}{2} w_j^2 + \log q(\mathbf{w}) \big] d\mathbf{w} + \text{const.} \tag{3.34}$$

Where $\langle z \rangle$ denotes an expectation. Since $\int q \log(p/q) dq$ is maximized by setting $q = p$, we have that

$$q_w^* \propto \prod^{n} \sigma(y_i \mathbf{K}_i \mathbf{w}) \exp\big(-\sum^{k} \frac{\langle \alpha_j \rangle}{2} w_j^2 \big) \tag{3.35}$$

This cannot be optimized analytically, so to keep standard distributions, Bishop recommends using the variational bound introduced by Jaakola and Jordan (Jaakkola et Jordan, 1997), in which the resulting form is quadratic in $\mathbf{w}$, and so gives Gaussian $q^*(w)$. This is not straightforward, so I re-present this here.

The basis of the variational bound on the logistic is that the tangent at any point on the convex function is lower bound. Thus for any differentiable convex function f, letting $g(x) = f(\xi) + \frac{\partial}{\partial \xi} f\big|_{\xi} (x - \xi)$, is an approximate lower bound, with equality at $x = \xi$. Jaakola and Jordan begin by noting that

$$\log \sigma(x) = -\log(1+\exp(-x)) = \frac{x}{2} - \log(1+\exp(-x)) - \frac{x}{2} \tag{3.36}$$
$$= \frac{x}{2} - \log(1+\exp(-x)) - \log\exp\big(\frac{x}{2}\big) \tag{3.37}$$
$$= \frac{x}{2} - \log\big(\exp(\frac{x}{2}) - \exp(-\frac{x}{2})\big) \tag{3.38}$$

Here, $f(x) = -\log(\exp(\frac{x}{2}) - \exp(\frac{-x}{2}))$ is convex in x^2, which can be verified by taking derivatives. Now we apply the trick mentioned above, so that we have a lower bound on f. Note here that equality holds when $\xi^2 = x^2$.

$$f(x) \geq f(\xi) + \frac{\partial f(\xi)}{\partial \xi^2}(x^2 - \xi^2) \tag{3.39}$$
$$= f(\xi) + \frac{1}{4\xi}\tanh(\xi/2)(x^2 - \xi^2) \tag{3.40}$$
$$\tag{3.41}$$

So that, replacing $f(x)$ in (3.38) and exponentiating we have

$$\log \sigma(x) \geq \log \sigma(\xi) - \frac{x+\xi}{2} + \frac{1}{4\xi}\tanh(\xi/2)(x^2 - \xi^2) \tag{3.42}$$
$$\sigma(y_i \mathbf{K}_i \mathbf{w}) \geq \sigma(\xi_i)\exp\big(\frac{y_i \mathbf{K}_i \mathbf{w} - \xi_i}{2} - \lambda(\xi_i)((y_i \mathbf{K}_i \mathbf{w})^2 - \xi^2)\big) \tag{3.43}$$

Where $\lambda(\xi) = \frac{1}{4\xi}\tanh(\xi/2)$.

Now, replacing $p(y_i|\mathbf{K}_i\mathbf{w})$ with $g_i(y_i, \mathbf{K}_i, \mathbf{w}, \xi_i)$, $p(\mathbf{y}|\mathbf{K}, \mathbf{w}) \geq G(\mathbf{y}, \mathbf{K}, \mathbf{w}, \xi) = \prod^n g_i$ and a new error function $\mathcal{E}_G$, which bounds the original error function from above,

$$\mathcal{E}_G = -\int_w \int_\alpha q(\mathbf{w})q(\alpha)\log\frac{G(\mathbf{y}, \mathbf{K}, \mathbf{w}, \xi)\prod N(w_j, \alpha_j)\Gamma(\alpha_j|a, b)}{q(\mathbf{w})q(\alpha)} d\mathbf{w}d\alpha \geq \mathcal{E} \tag{3.44}$$

The optimal $\log q^*$ is given by maximizing

$$\int_w q(\mathbf{w})\big[-\sum^n \frac{y_i \mathbf{K}_i \mathbf{w}}{2} + \lambda(\xi_i)(y_i \mathbf{K}_i \mathbf{w})^2 + \sum^k \frac{\langle \alpha_j \rangle}{2} w_j^2 + \log q(\mathbf{w})\big] d\mathbf{w} + \text{const.} \tag{3.45}$$

And, expanding further,

$$\log q^*(\mathbf{w}) = \sum^{n} \frac{y_i \mathbf{K}_i \mathbf{w}}{2} - \lambda(\xi_i)(y_i \mathbf{K}_i \mathbf{w})^2 - \sum^{k} \frac{\langle \alpha_j \rangle}{2} w_j^2 \tag{3.46}$$

$$= \frac{-1}{2}(-\sum y_i \mathbf{K}_i \mathbf{w} + 2\mathbf{w}^T \mathbf{K}^T \langle \Lambda(\xi) \rangle \mathbf{K}\mathbf{w} + \mathbf{w}^T \langle \mathbf{A} \rangle \mathbf{w}) \tag{3.47}$$

where $\mathbf{A} = diag(\langle \alpha \rangle)$, and $\Lambda = diag(\lambda(\xi))$. The resulting factorized form, after exponentiating is a Gaussian, $q^*(\mathbf{w}) = N(\mathbf{w}|\mathbf{m}, \mathbf{S})$. The moments are therefore

$$\mathbf{S} = (\mathbf{A} + 2\mathbf{K}^T \Lambda \mathbf{K})^{-1} \tag{3.48}$$

$$\mathbf{m} = \frac{1}{2}\mathbf{S}\big(\sum y_i \mathbf{K}_i\big); \tag{3.49}$$

The factorized distribution with respect to α are given by

$$\int_{\alpha} q(\alpha)\big[-\frac{1}{2}\sum \log \alpha_j + \frac{1}{2}\alpha_j \langle w_j^2 \rangle - \sum (a-1)\log(\alpha_j) + b\alpha_j \log q(\alpha)\big] \tag{3.50}$$

So that

$$\log q(\alpha) = \frac{1}{2}\sum \log \alpha_j - \sum \frac{1}{2}\alpha_j \langle w_j^2 \rangle + (a-1)\sum \log(\alpha_j) - b\sum \alpha_j + \text{const.} \tag{3.51}$$

$$= \big(a + \frac{1}{2} - 1\big)\sum \log \alpha_j - \sum \alpha_j \big(\frac{1}{2}\langle w_j^2 \rangle + b\big) + \text{const.} \tag{3.52}$$

Exponentiating,

$$q(\alpha) \propto \prod \alpha_j^{(a+\frac{1}{2}-1)} \exp\big(-\alpha_j(\frac{1}{2}\langle w_j^2 \rangle + b)\big), \tag{3.53}$$

which is an unnormalized Gamma distribution, $Gamma(\alpha_j|\hat{a}, \hat{b}_j)$, with $\hat{a} = a + \frac{1}{2}$, and $\hat{b}_j = \frac{1}{2}\sum \langle w_j^2 \rangle + b$.

Finally, noting that equality in the bound is given when $\xi_i^2 = (y_i \mathbf{K}_i \mathbf{w})^2$, $\xi_i = \mathbf{K}_i \langle \mathbf{w}\mathbf{w}^T \rangle \mathbf{K}_i^T$.

The algorithm is therefore a generalized EM procedure. With ψ as the digamma function, it updates $\mathbf{w}$, α and ξ, according to the following :

$$\mathbf{S} = (\langle \mathbf{A} \rangle + 2\mathbf{K}^T \Lambda \mathbf{K})^{-1} \tag{3.54}$$

$$\langle \mathbf{w} \rangle = \frac{1}{2} \mathbf{S} \sum y_i \mathbf{K}_i^T \tag{3.55}$$

$$\langle \mathbf{w}\mathbf{w}^T \rangle = \mathbf{S} + \langle \mathbf{w} \rangle \langle \mathbf{w} \rangle^T \tag{3.56}$$

$$\hat{a} = a + \frac{1}{2} \tag{3.57}$$

$$\hat{b_j} = \frac{1}{2} \sum \langle w_j^2 \rangle + b \tag{3.58}$$

$$\langle \alpha_j \rangle = \frac{\hat{a}}{\hat{b_j}} \tag{3.59}$$

$$\langle \log \alpha_j \rangle = \psi(\hat{a}) - \log \hat{b_j} \tag{3.60}$$

$$\xi_i = \mathbf{K}_i \langle \mathbf{w}\mathbf{w}^T \rangle \mathbf{K}_i^T. \tag{3.61}$$

In order to test for convergence, compute the current $\mathcal{E}$,

$$\begin{aligned} \mathcal{E}_G = &-\langle \log f(y, K, \mathbf{w}, \xi) \rangle - \langle \log p(\mathbf{w}|\alpha) \rangle \\ &- \langle \log p(\alpha) \rangle + \langle \log q(\mathbf{w}) \rangle + \langle \log q(\alpha) \rangle \end{aligned} \tag{3.62}$$

$$\begin{aligned} \langle \log f(y, K, \mathbf{w}, \xi) \rangle = &\sum \log \sigma(\xi_i) - \frac{1}{2}(y_i \mathbf{K}_i \langle \mathbf{w} \rangle - \xi_i) \\ &\sum -\lambda(\xi_i)(\frac{1}{2} y_i \mathbf{K}_i \langle \mathbf{w} \rangle)(\mathbf{K}_i \langle \mathbf{w}\mathbf{w}^T \mathbf{K}_i^T \mathbf{w} \rangle - \xi_i^2) \end{aligned} \tag{3.63}$$

$$\langle \log p(\mathbf{w}|\alpha) \rangle = \frac{n}{2} \log(2\pi) + \frac{1}{2} \sum \langle \log \alpha_j \rangle - \frac{1}{2} \langle w_j^2 \rangle \langle \alpha_j \rangle \tag{3.64}$$

$$\langle \log p(\alpha) \rangle = \sum a \log b + (a-1) \langle \log \alpha_j \rangle - b \langle \alpha_j \rangle - \log \Gamma(a) \tag{3.65}$$

$$\langle \log q(\mathbf{w}) \rangle = -n/2(1 + \log(2\pi)) - 1/2 \log |\mathbf{S}| \tag{3.66}$$

$$\langle \log q(\alpha) \rangle = \sum \hat{a} \log b + (\hat{a} - 1) \langle \log \alpha_j - \hat{b} \langle \alpha_j \rangle - \log \Gamma(\hat{a}). \tag{3.67}$$

Although the predictive distribution should be taken as

$$\int \int p(y|\mathbf{w}) p(\mathbf{w}|\alpha) d\mathbf{w} d\alpha \tag{3.68}$$

This integral is intractable, and so the predictive distribution is approximated with the mean $\langle \mathbf{w} \rangle$,

$$p(y|\langle \mathbf{w} \rangle) \tag{3.69}$$

This preserves the original intent of the RVM, which is to produce fast classifiers. Although the lower bound is guaranteed to converge, there are numerous local minimum and so initialization is quite important. In practice, the initialization of α is quite important and several non-intuitive results are shown below. The presence of these local minimum as a function of the hyperpriors make the variational RVM less attractive than the fast evidence approximating RVM, which uses an implicit Jeffrey's prior and MAP estimation.

In figure (3.6), the results of initializing p percent of the α to large values is shown, effectively acting as a prior. Sparsity is monotonically increasing, with the results being shown averaged over 10 runs. The trickiness of initialization is also fascinating, as setting an entire subset of α to a particular value has interesting consequences. For instance, our beliefs that only α that are associated with one class "matter" can be encoded by initialization. The last row in figure (3.6) shows an average over 10 runs where half the α, for which $(y_i = -1)$ were in fact set to high values.

Table 3.6 Some tabulated results for Variational RVM, details are in text.

	training error	testing error	Bayes error	NSV	p
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
VRVM	.142	.200	.19	147	-
VRVM	.136	.187	.19	145	25
VRVM	.157	.211	.19	75	30
VRVM	.168	.254	.19	7	50
VRVM-pos	.167	.233	.19	7	50

One of these runs is depicted in figure (3.7), where clearly all the relevance vectors are from class +1. How to leverage/avoid this behavior is a direction for further research.

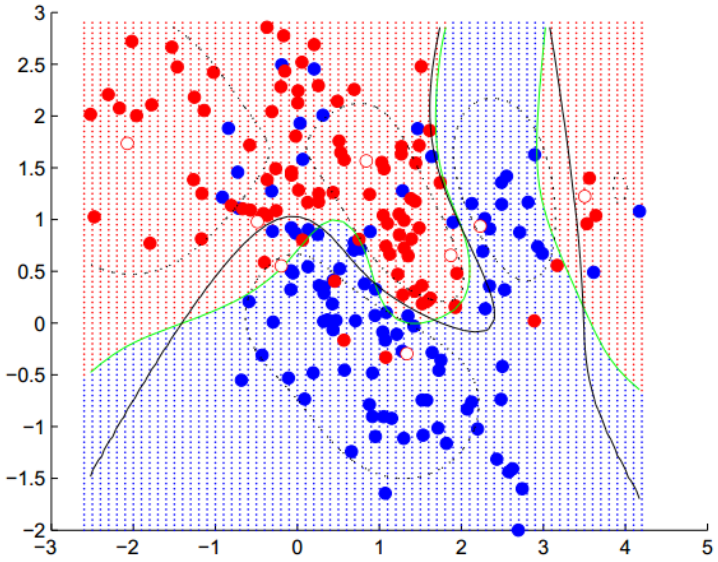


Figure 3.7 Variational RVM : Only positive relevance vectors are used. Details in are in the text.

3.2 Summary of Results

Table 3.7 Summary

	training error	testing error	Bayes error	NSV	params
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
Laplace	.150	.235	.19	198	.0064
Exponential	.155	**.200**	.19	55	.1024
Jeffrey's	.155	.225	.19	**7**	-
Gen. Gaussian	.160	**.200**	.19	15	p=.1,.4096
VRVM	**.142**	**.200**	.19	147	-

In conclusion, the "l_1" priors are convex, but not necessarily sparse. At the other extreme the variational RVM *can* reach this globally maximally sparse solution, but the presence of many local solutions dependent on the initialization makes the variational RVM less attractive. However, the power of the variational RVM may possibly be harnessed by viewing the initial state as a prior encoding domain knowledge.

Figure (3.7) contains a summary of each prior, where RVM denotes the RVM based on the Laplace approximation first presented by Tipping and Faul, (Faul et Tipping, 2002). The

Generalized Gaussian and Exponential, however, provide best results on this dataset. However, the Exponential has the interesting combination of properties of l_1 and l_2 regularization, suggesting a good trade-off between sparsity and shrinkage. Because of this, the Exponential prior appears to be promising for further research.

3.3 Multi-class Variational Bayes with Exponential

Adapting the Exponential model to the multi-class case is straight-forward. The form of the multinomial distribution implied by the logistic distribution leads to the "soft-max" formulation. In this case, y_n becomes a binary vector with the c^{th} element of $\mathbf{y}_n$ equal to 1, denoting the class of the n^{th} label, and 0 elsewhere. A background class can be chosen so that M classes can be represented with $C = M - 1$ dimensions :

$$\mathbf{W} = \text{argmax}_{\mathbf{W}} \log \prod_n^N p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}) = \text{argmax}_{\mathbf{W}} \sum_n^N \log \frac{\exp(\mathbf{y}_n^T \mathbf{W}\phi(\mathbf{x}_n))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_n))}. \quad (3.70)$$

The prior adds an additional term to the objective, as in

$$\mathbf{W} = \text{argmax}_{\mathbf{W}} \log \prod_n^N p(\mathbf{W}|\mathbf{y}_n, \mathbf{x}_n) \propto \text{argmax}_{\mathbf{W}} \log \prod_n^N p(\mathbf{y}_n|\phi(\mathbf{x}_n), \mathbf{W}) p(\mathbf{W}) \quad (3.71)$$

with

$$\mathbf{W} = \text{argmax}_{\mathbf{W}} \log \prod_n^N p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}) p(\mathbf{W}) \quad (3.72)$$

$$= \text{argmax}_{\mathbf{W}} \sum_n^N \log \frac{\exp(\mathbf{y}_n^T \mathbf{W}\phi(\mathbf{x}_n))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_n))} - \frac{1}{2} \sum_c^C \mathbf{W}_c^T \mathbf{A}^{-1} \mathbf{W}_c, \quad (3.73)$$

where $\mathbf{W}_c$ refers to the c^{th} row of the $\mathbf{W}$ matrix and covariance matrix $\mathbf{A}^{-1}$ set as a hyper-parameter, usually diagonal and isotropic. As shown in Figure 3.1(b), the inclusion of this variable yields the joint distribution

$$p(\mathbf{Y}, \mathbf{A}, \mathbf{W}|\mathbf{X}) = \prod_n^N p(\mathbf{y}_n|\phi(\mathbf{x}_n), \mathbf{W}) p(\mathbf{W}|\mathbf{A}) p(\mathbf{A}) \quad (3.74)$$

That is, let $\mathbf{A}$ be diagonal but non-isotropic. This implies that $\mathbf{A}$ can be represented as

a vector of NC hyper-parameters α_{nc}, and let

$$p(w_{nc}) = \mathcal{N}(w_{nc}|0, \alpha_{nc}) \tag{3.75}$$

$$p(\alpha_{nc}) = \text{Exp}(\frac{\gamma}{2}), \tag{3.76}$$

for all N examples and all classes C. This results again in an Exponential prior, which approximates an L_1 penalty, as opposed to an L_2 penalty, which will not generally yield sparse $\mathbf{W}$, but provides a shrinkage effect (Krishnapuram *et al.*, 2005). The issue is that now the posterior distribution has an integral

$$p(\mathbf{W}|\mathbf{y}, \mathbf{X}) \propto \prod_n^N p(\mathbf{y}_n|\phi(\mathbf{x}_n), \mathbf{W}) \int_{\mathbf{A}} p(\mathbf{W}|\mathbf{A})p(\mathbf{A})d\mathbf{A}, \tag{3.77}$$

so that the usual MAP procedure implies taking the arg max of an objective function which contains the log of a summation, as now

$$\begin{aligned}\mathbf{W} = \text{argmax}_{\mathbf{W}} \log \prod_n^N \frac{\exp(\mathbf{y}_n^T \mathbf{W}\phi(\mathbf{x}_n))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_n))} \\ \prod_{nc} \int_{\alpha_{nc}} \left(\mathcal{N}(w_{nc}|\alpha_{nc}) \frac{\gamma}{2} \exp(-\frac{\gamma}{2}\alpha_{nc}) \right) d\alpha_{nc}.\end{aligned} \tag{3.78}$$

However, as shown in (Tipping, 2000a), we can optimize an alternative objective which is a lower-bound on the above. In this case, we are primarily interested in optimizing the conditional probability of the labels $\mathbf{Y}$, given the feature inputs $\mathbf{X}$. The joint distribution

$$p(\mathbf{Y}, \mathbf{W}, \mathbf{A}|\mathbf{X}) = \prod_n^N \frac{\exp(\mathbf{y}_n^T \mathbf{W}\phi(\mathbf{x}_n))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_n))} \prod_{nc} \left(\mathcal{N}(w_{nc}|\alpha_{nc}) \frac{\gamma}{2} \exp(-\frac{\gamma}{2}\alpha_{nc}) \right). \tag{3.79}$$

Given some distribution $q(\mathbf{W}, \mathbf{A})$, we can show that

$$\log p(\mathbf{Y}|\mathbf{X}) = \int_{\mathbf{W},\mathbf{A}} q(\mathbf{W}, \mathbf{A}) \log \left(\frac{p(\mathbf{Y}, \mathbf{W}, \mathbf{A}|\mathbf{X})}{q(\mathbf{W}, \mathbf{A})} \right) d\mathbf{W}d\mathbf{A} + KL(q||p). \tag{3.80}$$

Because $KL(q||p)$ is non-negative, the first quantity in Equation (3.80), which we can refer to as $\mathcal{L}(q)$, is a lower bound on the objective

$$\int_{\mathbf{W},\mathbf{A}} q(\mathbf{W}, \mathbf{A}) \log \left(\frac{p(\mathbf{Y}, \mathbf{W}, \mathbf{A}|\mathbf{X})}{q(\mathbf{W}, \mathbf{A})} \right) d\mathbf{W}d\mathbf{A} = \mathcal{L}(q) \leq \log p(\mathbf{Y}|\mathbf{X}). \tag{3.81}$$

Another way to write our objective, as an expectation, is

$$\mathcal{L}(q) = \mathbb{E}_{q(\mathbf{W},\mathbf{A})}[\log p(\mathbf{Y}, \mathbf{W}, \mathbf{A}|\mathbf{X})] + \mathbb{H}[\mathbf{W}, \mathbf{A}] \tag{3.82}$$

$$= \mathbb{E}_{q(\mathbf{W},\mathbf{A})}[\log p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W}|\mathbf{A})p(\mathbf{A})] + \mathbb{H}[\mathbf{W}, \mathbf{A}] \tag{3.83}$$

where $\mathbb{H}$ is the entropy of the q distribution. In words, we wish to maximize the expected log joint probability under the q distribution, regularized by the entropy of the q distribution. Using variational inference (Attias, 2000), in what (Ghahramani et Beal, 2001) term the variational Bayesian approach, we let the distribution factorize, so that

$$q(\mathbf{W}, \mathbf{A}) = q(\mathbf{W})q(\mathbf{A}) = \prod_{nc} q(w_{nc})q(\alpha_{nc}). \tag{3.84}$$

Our objective, therefore, can now be written with these terms,

$$\mathcal{L}(q) = \mathbb{E}_{q(\mathbf{W})}[\sum_{n}^{N} \log \frac{\exp(\mathbf{y}_n^T \mathbf{W}\phi(\mathbf{x}_n))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_n))}] - \frac{1}{2}\sum_{nc} \mathbb{E}_{q(w_{nc},\alpha_{nc})}[\log(\alpha_{nc}) + \alpha_{nc}^{-1} w_{nc}^2 + \gamma\alpha_{nc}] + \text{const.} \tag{3.85}$$

The only thing remaining is to specify the distributions, $q(\alpha_{nc})$ and $q(w_{nc})$. In our model, the distribution of α_{nc} is log proportional to the joint, and similarly ignoring constant terms, using the $\langle\dot{}\rangle$ notation to denote an expectation, we have

$$q(\alpha_{nc}) \propto \langle \alpha_{nc}^{-\frac{1}{2}} \exp(-\frac{1}{2}(\gamma\alpha_{nc} + w_{nc}^2\alpha_{nc}^{-1}))\rangle_{q(w_{nc})}. \tag{3.86}$$

We have chosen $q(\mathbf{w})$ to simplify things greatly. Changing variables, let $\tau_{nc} = \alpha_{nc}^{-1}$, resulting in the distribution

$$q(\tau_{nc}) \propto \tau_{nc}^{-\frac{3}{2}} \exp(-\frac{1}{2}(\gamma\tau_{nc}^{-1} + \langle w_{nc}^2\rangle_{q(w_{nc})}\tau_{nc}), \tag{3.87}$$

$$\mathcal{L}(q) = \int q(\alpha)\Big(-\frac{1}{2}\sum_{nc} \log(\alpha_{nc}) - \sum_{nc} \frac{1}{2}\alpha_{nc}^{-1} w_{nc}^2 - \frac{\gamma}{2}\sum_{nc} \alpha_{nc}\Big) d\alpha - \mathbb{H}[q(\alpha)] + \text{const.}$$

Since this term is the negative KL divergence, which is non-positive, the term is maximized (the KL divergence is minimized), when the value is 0. That is,

$$\log q(\alpha) \propto \Big(-\frac{1}{2}\sum_{nc} \log(\alpha_{nc}) - \sum_{nc} \frac{1}{2}\alpha_{nc}^{-1} w_{nc}^2 - \frac{\gamma}{2}\sum_{nc} \alpha_{nc}\Big)$$

Changing variables with $\tau_{nc} = \alpha_{nc}^{-1}$, using $|\frac{\partial}{\partial \alpha_{nc}} \tau_{nc}| = \frac{1}{\alpha^2}$, and some simplifying,

$$\begin{aligned} q(\tau_{nc}) &\propto \tau_{nc}^{-3/2} \exp(-\frac{1}{2}(\frac{\gamma}{\tau_{nc}} + w_{nc}^2 \tau_{nc})) \\ &\propto \frac{1}{\sqrt{\tau_{nc}^3}} \exp(-\frac{\gamma}{2}(\frac{1}{\tau_{nc}} + \frac{w_{nc}^2}{\gamma} \tau_{nc})) \\ &\propto \frac{1}{\sqrt{\tau_{nc}^3}} \exp(\frac{-\gamma(\tau_{nc} - \sqrt{\gamma}|w_{nc}|^{-1})}{2\tau_{nc}\gamma w_{nc}^{-2}}) \end{aligned}$$

This has the form of the inverse Gaussian distribution with mean $\frac{\sqrt{\gamma}}{|w_{nc}|}$, and shape parameter γ (Chhikara et Folks, 1989a). We note that this is not the true posterior, as we do not use a variational or posterior distribution for $\mathbf{w}$, but are using the maximum likelihood estimate of $\mathbf{w}$ according to the model and the prior distribution of w. In effect, this can be interpreted as a mode of the implied posterior distribution of $\mathbf{w}$.

For $q(\mathbf{W})$, we treat the rows of $\mathbf{w}_c$ as equivalent to a point distribution on the current mode given by maximization, $q(\mathbf{W}) = \delta(\mathbf{W} = \mathbf{W}^k)$. The E-step for $q(\mathbf{W})$ is then the maximization of the objective with respect to the q distribution

$$\begin{aligned} \nabla_{\mathbf{w}_c} \mathcal{L}(q) = \sum_n y_{nc} \phi(\mathbf{x}_n) - \sum_n \frac{\exp(\mathbf{y}_n^T \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_n))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_n))} \phi(\mathbf{x}_n) \\ + \langle \mathrm{T}_c \rangle_{q(\mathrm{T}_c)} \mathbf{w}_c, \end{aligned} \tag{3.88}$$

where $\langle \mathrm{T}_c \rangle_{q(\mathrm{T}_c)}$ is the diagonal matrix with $E_{q(\tau_{nc})}[\tau_{nc}]$ along the diagonal. To summarize, we iterate between obtaining the current estimates, with ϵ as a step-size,

$$\langle \mathbf{w}_c \rangle_{q(\mathbf{w}_c)} = \mathbf{w}_c^k - \epsilon \nabla_{\mathbf{w}_c} \mathcal{L}(q) \tag{3.89}$$

$$\langle \tau_{nc} \rangle_{q(\tau_{nc})} = \frac{\sqrt{\gamma}}{|\langle w_{nc} \rangle_{q(w_{nc})}|}. \tag{3.90}$$

As w_{nc} gets close to 0, the penalty approaches infinity, generating sparse solutions. We choose a sufficiently large constant to model infinity. We also monitor convergence by approximating the objective function Equation (3.85), with

$$\begin{aligned} \tilde{\mathcal{L}}(q) = \sum_n^N \log \frac{\exp(\mathbf{y}_n^T \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_n))}{\sum_{c=1}^C \exp(\mathbf{1^{(c)}} \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_n))} \\ - \frac{1}{2} \sum_{nc} \log(\langle \tau_{nc}^{-1} \rangle_{q(\tau_{nc})} + \langle \tau_{nc} w_{nc}^2 \rangle_{q(\tau_n c)} + \gamma \langle \tau_{nc} \rangle_{q(\tau_{nc})}^{-1} + \text{const.} \end{aligned} \tag{3.91}$$

3.4 Structured Prediction with Sparse Kernel Priors : A Relevance Vector Random Field

The impressive performance of kernel methods such as the Support Vector Machine (SVM) (Cortes et Vapnik, 1995; Weston et Watkins, 1999) and Kernel Logistic Regression (KLR) (Jaakkola et Haussler, 1999), (Zhu et Hastie, 2001) have lead to much interest in their use in structured settings. Structured models based on the SVM hinge-loss and maximum margin approach have been presented by (Taskar *et al.*, 2003) as the Max-Margin Markov Network (M3N), and (Tsochantaridis *et al.*, 2005), in a more general approach with similar objectives known as the Structured SVM. The structured extension of the KLR is best represented by the Kernel Conditional Random Field (KCRF) (Lafferty *et al.*, 2004). In both cases, the feature functions ρ are replaced by kernel functions K and regularized using the L_2 penalty.

The main difference between SVM-based structured models and the kernel CRF is the loss function, as the log-likelihood loss in the kernel CRF is replaced by the hinge loss function of the SVM. The main benefit of the choice of loss function is that it leads to a sparse solution, more easily seen in the dual formulation of the SVM. The KCRF, on the other hand, requires alternative measures to ensure sparse solutions. In a full KCRF, the number of parameters can be very large as there is no restriction on the number of basis functions, typically the size of the entire training set. In (Lafferty *et al.*, 2004), a greedy iterative approach is used to reduce the required number of kernel functions.

However, the main benefit of the KCRF is that, as a fully probabilistic model, it allows for the use of standard probabilistic modeling tools. Learning and inference, therefore, can be accomplished using methods used in any CRF formulation. However, it requires many iterations to reach an optimal solution. Meanwhile, the M3N optimization is a Quadratic Programming (QP) problem that is usually very large in practice and therefore slow or sometimes impossible to solve. In both cases inexact methods are usually required.

As the M3N derives from the SVM and CRF, the KCRF can be seen as a structured version of the Import Vector Machine (IVM) (Zhu et Hastie, 2001). The IVM is a KLR method which uses a greedy approach for kernel basis selection, in a manner similar to stepwise regression (Hocking, 1976). However, sparse methods such as the Relevance Vector Machine (RVM) can also be used to derive sparse KLR solutions (Tipping, 2000b). In the RVM, a Bayesian approach which uses a carefully chosen prior results in the desired sparsity. However, the Automatic Relevance Determination prior used in (Tipping, 2000b) requires an approximate Hessian, which leads to it's own complexity issues.

A "fast" version of the RVM was presented which did not require computation of a full Hessian was presented in (Tipping *et al.*, 2003), but resembles the IVM in that it uses a

selection set chosen by a scoring function. Although the scoring function is more principled, it requires multiple solutions, bounded by the size of the selected set of basis functions, to a KLR problem that is already quadratic.

As shown above, however, rather than full Bayesian inference, the use of an Exponential hyperprior for the variance results in an Exponential prior, which approximates an L_1 penalty, as opposed to an L_2 penalty which will not generally yield sparse solutions. The resulting prior provides shrinkage and sparsity, approximating the penalty of the RVM in a MAP setting.

The extension of the prior to the CRF model in this paper is straight-forward in that the components of the method are broken into two alternating steps. First, the prior parameters (the variance of the prior) are updated and the weights of the local and interaction potentials are updated based on the updated variance parameter and choice of inference procedure.

3.4.1 Learning, Inference and Approximations

Learning can be more formally defined as an optimization procedure for finding the parameters of a given model. For learning in MRFs, there are two broad categories of techniques : those based on maximum likelihood (ML) and those based on variations of the maximum margin (MM) approach made popular by SVMs. Given labels $\mathbf{y}$ and features $\mathbf{x}$ in ML, the CRF optimization objective is given by the log conditional probability of labels given features $\log p(\mathbf{y}|\mathbf{x})$. In these typical ML learning settings, learning based on some form of gradient descent will lead to computation for gradients that takes the following form :

$$\frac{\partial}{\partial\theta}\log p(\mathbf{y}|\mathbf{x};\theta) \propto \Big\langle \frac{\partial\rho(\mathbf{x},\mathbf{y};\theta)}{\partial\theta} \Big\rangle_{\tilde{p}(\mathbf{x},\mathbf{y})} - \Big\langle \Big\langle \frac{\partial\rho(\mathbf{x},\mathbf{y};\theta)}{\partial\theta} \Big\rangle_{p(\mathbf{y}|\mathbf{x};\theta)} \Big\rangle_{\tilde{p}(\mathbf{x})}, \tag{3.92}$$

where ρ represents the feature function in the exponent of the MRF model. The gradient can be interpreted as the difference between an empirical expectation and (a more complicated) model expectation. Using the above gradient the parameters can be updated at each iteration of the gradient descent step as follows :

$$\theta_k = \theta_k^{old} - \eta\frac{\partial L}{\partial\theta_k}$$

where η is a learning rate.

The CRF model can be extended with the Exponential hyperprior and kernel functions to produce a sparse kernel CRF. First, ρ is now replaced with a function f_K dependent on a

kernel function K. The notation of the objective function is only slightly altered by using a kernel representation,

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{p} \exp\Big(- \sum_{c=1}^{C} \sum_{m=1}^{M} \lambda_{c,m} f_K(x_p^{int}, x_m^{int}, y_p) \Big) \prod_{p,q \in N_{xy}} \exp\Big(- \gamma \rho_\gamma(y_p, y_q, x_{pq}^{grad}) \Big) \prod_{p,q \in N_z} \exp\Big(- \theta \rho_\theta(y_p, y_q, x_{pq}^{grad}) \Big), \tag{3.93}$$

where the function $f_K(x_p, x_m, y_p) = \mathbf{1}_{y_p=c} K(x_p, x_m)$ uses the RBF kernel instead of binary feature functions and is defined as follows, $K(a, b) = \exp(-\frac{1}{2\sigma^2}||a - b||^2)$. This parameterization can be interpreted as a "smoothed" histogram whose bandwidth is given by σ. The indicator function $\mathbf{1}_{y=c}$, which is defined as 1 when $y = c$ and 0 otherwise, encodes the dependence on the label output. M denotes the number of "relevant" vectors and x_m denotes the relevance vector. Note that c is passed implicitly to the function f_K. The range of intensity values is small and discrete, while the total number of pixels per patient is quite large. Therefore only a small number of intensity values should be required for use as kernel basis functions.

In order to introduce sparsity into the model, a Gaussian prior for the parameters λ and an Exponential hyperprior on the variance of the Gaussian is added, resulting in the joint distribution

$$p(\mathbf{y}, \lambda|\mathbf{x}) \propto p(\mathbf{y}|\mathbf{x}, \lambda) \prod_{c,m} p(\lambda_{c,m}|\alpha_{c,m}) p(\alpha_{c,m}), \tag{3.94}$$

where $p(\lambda_{c,m}|\alpha_{c,m}) \sim N(0, \alpha_{c,m})$ and $\alpha_{c,m} \sim \text{Exp}(\frac{\kappa}{2})$. The result of this hierarchical model, after integrating out $\alpha_{c,m}$, is an Exponential prior

$$p(\lambda_{c,m}) = \sqrt{\kappa} 2 \exp(-\frac{\sqrt{\kappa}}{2} |\lambda_{c,m}|). \tag{3.95}$$

If we view $p(\mathbf{y})$ as delta function $\delta(\mathbf{y} - \hat{\mathbf{y}})$, where $\hat{\mathbf{y}}$ is the maximizing value or mode,

$$p(\alpha|\lambda, \kappa, \mathbf{y}, \mathbf{x}) \propto \prod_{m} \prod_{c} \exp\Big(- \frac{1}{2\alpha_{c,m}} \lambda_{c,m}^2 - \frac{\kappa}{2} |\alpha_{c,m}| \Big). \tag{3.96}$$

This has the form of the inverse Gaussian distribution with mean $\frac{|\lambda_{c,m}|}{\sqrt{\kappa}}$, and shape parameter

$\frac{1}{\kappa}$ (Chhikara et Folks, 1989b). Therefore this posterior can again be used in a Variational Bayesian procedure where λ is treated as a parameter, optimizing the EM objective

$$\mathcal{L}(p) = \int p(\alpha|\lambda,\kappa)\log p(\mathbf{y},\mathbf{x},\lambda) - \mathbb{H}[p(\alpha|\lambda,\kappa)], \tag{3.97}$$

where $\mathbb{H}[]$ denotes the entropy of the posterior. Therefore, the expectation step consists of the update

$$\alpha_{m,c} = \frac{\sqrt{\kappa}}{|\lambda_{c,m}|}, \tag{3.98}$$

and the updates to λ and γ can be updated using gradient descent as usual. However, the gradient now incorporates an additional penalty term

$$\begin{aligned}\frac{\partial L_{CRF}}{\partial \lambda_{c,m}} &= \sum_{p} f_K(\mathbf{x_n},\mathbf{x_m},\mathbf{y_p}) - \sum_{\mathbf{y}'} p(\mathbf{y}'|\mathbf{x}) f_K(\mathbf{x_m},\mathbf{x_n},\mathbf{y}') \\ &\quad - \frac{\alpha_{c,m}}{2}\lambda_{c,m}.\end{aligned} \tag{3.99}$$

Note that as $|\lambda_{c,m}|$ gets small, $\frac{\alpha_{c,m}}{2}$ becomes large as well, forcing λ towards 0. This novel formulation of a sparse kernel CRF can be viewed as a relevance vector random field (RVRF). More details and portions of this section appears in (Bhole *et al.*, 2013)

3.5 Experiments and Results

One important application area of the CRF framework is for segmentation, (Kumar et Hebert, 2006), (Plath *et al.*, 2009), (Reynolds et Murphy, 2007), (Hoiem *et al.*, 2007). CRF segmentation has also been applied in the medical imaging field. In this section, we apply the sparse kernel CRF on a realistic problem, highlighting how the algorithm may be used in practice.

In experiments, five 3D CT volumes of patients from a 3D liver dataset are used. The number of slices containing the adrenal gland range from 15 slices to 52 slices. The segmentations of the clinical left adrenal gland are obtained from an abdominal expert radiologist. Windowed volumes that contain the adrenal gland are used. An example of the adrenal gland with its location in an axial slice is shown in Figure 3.8. The size of the gland varies and could be small and hence we use a windowed region for the experiments. The images at full resolution are used. The intensity values are chosen to range between -180 to 1200HU.

3 slices labeled by an expert radiologist are chosen as training for each patient. In addition to these 3 slices, 2 slices in the exterior of the adrenal gland volume one on each side since all pixels in these two slices are background. For CRF and our RVRF experiments for each

patient volume, 5 slices are used to train the model parameters. Testing is then performed on the whole patient volume containing the adrenal gland. The 3 slices used for training are used as constraints for segmentation in testing. Hence, the setup is somewhat interactive.

To compare our approach with established methods results using an implementation of an active contour model with a shape prior based on the work of (Bresson *et al.*, 2006) are shown.The adrenal gland changes shape dramatically between the first and last slice, so the interactive training examples are treated as unique examples and use the nearest interactive training slice to initialize the shape prior. The model has around 20 free parameters that the user needs to tune. One important parameter is the shape prior weighting term. It is necessary to set the shape prior weighting to a high value when the test slice is close to the interactive training slice and to a low value when they are far apart. Moreover, determining what values to select for what distance is not automated and needs to be selected manually.

Comparisons of segmentations for the same testing slices are shown in Fig. 3.8 in which the results of using the contour model approach (Bresson *et al.*, 2006) in the upper row and using the CRF in the lower row is shown. The same set up is shown in both cases, i.e. where 3 slices are used as interactive input. The red curves are the shape priors that are used in the contour model and which are modified to fit the new data in the new slice. It does not always settle to the correct outline of the adrenal gland. The green curves are the CRF segmentation results. Since the contour model needed to be fine tuned per frame (a common set of parameters did not work), the results for only two slices are shown.

The pixel accuracy on all the 5 patient volumes (test slices) using the CRF model is 95.22 % and the average class accuracy is 86.59 %. A 3D active contour model described in (Zhang *et al.*, 2008) where image gradients are used in their hybrid model and set the gradients of the interactive slices to hard values for fairness to make it similar for the data. A pixel accuracy of 92.96% and average class accuracy of 84.56% for all 5 volumes was obtained.

Two additional steps were necessary to lower computational requirements for the RVRF. First, sampling 25 percent of the number of possible kernel basis functions in order to reduce the time to compute kernel functions, yielding, on average, 4394 initial kernel basis functions. Secondly, optimizing the approach using pseudo-likelihood training to optimize both over all variables, and then perform the optimization separately using LBP for inference, in which α are fixed. This essentially reduces the candidate functions before full CRF training. The κ hyper parameter and σ parameter in the RBF kernel function are both set to 1 for all patients, chosen by optimizing over the training set. In our experiments, 62% reduction in basis functions on average after pseudo-likelihood training was obtianed. The pixel accuracy using this model is 97.60 % and average class accuracy improved to 87.20 %. Table 3.8 summarizes these results.

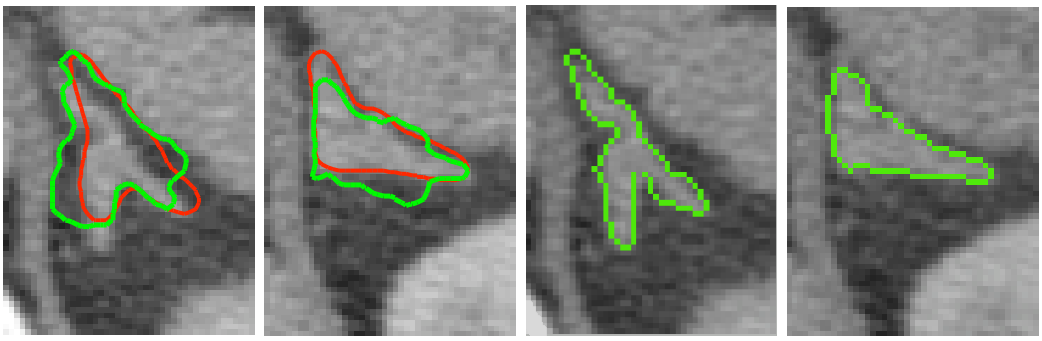


Figure 3.8 Segmentation examples using the model. Upper row shows results using a contour model obtaining a pixel accuracy of 90.73 and 94.31 for two different slices and the lower row using the CRF obtains a pixel accuracy of 96.72 and 97.48 respectively. The red is a modified shape prior and green is the segmentation result.

Table 3.8 This table summarizes the different accuracies for the adrenal segmentation problem using different models. ACA is the average class accuracy, PA is the pixel accuracy.

	3D active contour model	**CRF**	**Sparse kernel CRF**
ACA	84.56	86.59	87.20
PA	92.96	95.22	97.60

3.6 Discussion

As shown in the evaluation, the novel 3D sparse kernel CRF technique boosts the performance of the 3D CRF based approach while avoiding feature discretization. This technique has a lot of potential to provide state of the art results on other problems in a way that allows the system designer to avoid commonly used discretization, clustering or code-booking steps. As this approach combines key ideas from relevance vector machines (RVMs) and random fields, the framework is can be called relevance vector random fields or RVRFs.

CHAPTER 4

Face Recognition with Weakly Labeled Data

The previous chapter focused on general machine learning topics. Some of the methods in the previous chapter provide relevant background for this chapter, which moves toward facial image analysis. The second main theme of this dissertation is facial recognition. In particular, this chapter deals with facial recognition with weakly labeled data. Portions of this chapter appeared previously in (Rim *et al.*, 2011) and have been submitted for publication to Pattern Recognition.

Facial recognition research has recently focused on the difficult task of recognition in unconstrained images, *i.e.*photographs of faces under naturally occurring conditions. These images, usually produced by digital photography under conditions of varying illumination and pose, are often occluded (*e.g.*with glasses, body parts), and sometimes heavily compressed or degraded by motion blurring. Facial recognition remains a difficult task for these images. The availability of labeled examples is a primary source of this difficulty.

Finding sources of *unlabeled* facial images, however, is not a challenging task. Several large web-based collections such as Flickr and Google Images provide public access to millions of static images. More over, video sites such as YouTube provide access to videos which contain many more examples of human faces.

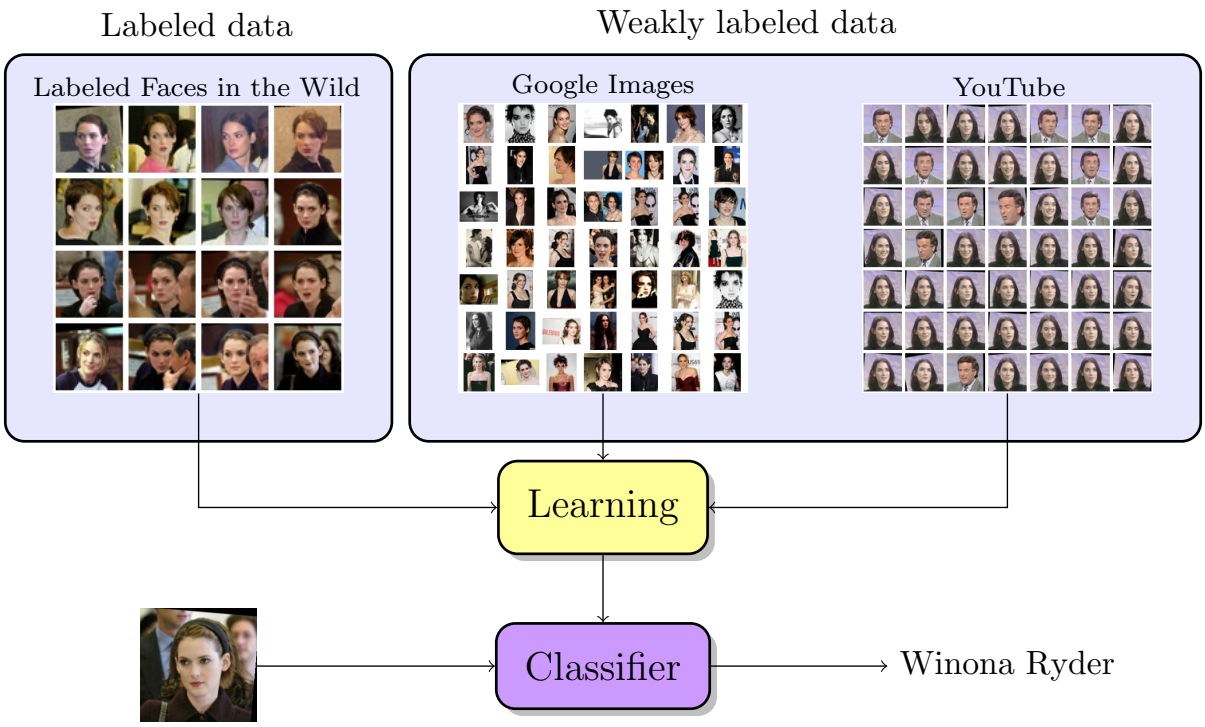


Figure 4.1 High performance recognition requires a large number of labeled examples in the uncontrolled case. Large amounts of weakly labeled data are easily obtainable through the use of image and video search tools. Many of these examples are either irrelevant or not the identity in question. We learn a classifier that does not require manual labeling of the weakly labeled data by accounting for the weak label noise.

Although the resulting data is unlabeled, useful information is retained and can be used as a "weak label." In this chapter, search queries are used as weak labels, in effect, by assuming the search engine is a good labeler. For unconstrained facial recognition in both static and video images using this information can significantly improve results. Figure (4.1) summarizes both the problem and our approach – relatively few labeled examples and many unlabeled examples.

4.1 Weakly Supervised Learning

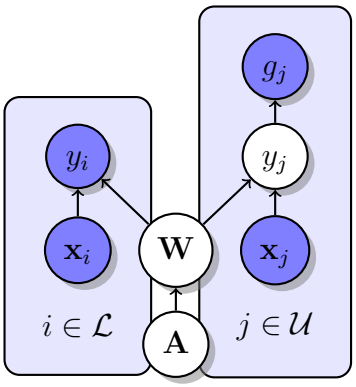


Figure 4.2 Graphical model shown in Figure 3.1(b) augmented for semi-supervised learning. $\mathcal{U}$ is the index set of unlabeled examples, and $\mathcal{L}$ is the index set of the labeled examples. g denotes noisy label variables for examples with unobserved labels y.

The general idea of this chapter is that the supervised model from the previous chapter results in a powerful classifier for use with labeled examples. However, because the model is fully probabilistic, the model can be adapted in a straight-forward way using the rules of probability.

The supervised model from the previous chapter, therefore, is here adapted into a model which allows for learning with weak labels. The graphical model shown in figure (4.2) represents the revised approach. The appropriate objective function emerges as a generalized Expectation-Maximization procedure adapted from a variational Bayesian perspective as described in the previous chapter, with only a few adaptations.

As shown in Figure 4.2, in contrast to the usual discriminative model, there is now additional information, encoded as the variable g, in which conditional dependence of g on y is assumed. Ideally, the variables g are random variables which are independent of $\mathbf{x}$ having observed y – once the identity of the subject in the video is known, no data from the image itself is necessary.

In this work, g is referred to as a weak or noisy label, as g should ideally be identical to y, making the problem "easy." In the labeled case, when the true label $\mathbf{Y}$ is observed, the additional variables become irrelevant, since $\mathbf{x}_n$ and $\mathbf{g}_n$ are conditionally independent given y_n. However, when $\mathbf{Y}$ is not observed, g becomes an important source of information.

Let $\mathcal{L}$ be an index set for the labeled data, and L the cardinality of the labeled set, *i.e.*$|\mathcal{L}|$. Let $\mathcal{U}$ be an index set for the unlabeled data, with U the cardinality of the labeled set, $|\mathcal{U}|$. The set $\{g_j\}_{j\in\mathcal{U}}$ is represented by $\mathbf{G}$, and the set of observed labels, $\{y_i\}_{i\in\mathcal{L}}$ by $\mathbf{Y}_l$, and the set of missing labels, $\{y_i\}_{i\in\mathcal{L}}$, by $\mathbf{Y}_u$.

The missing labels, $\mathbf{Y}_u$ is treated as hidden variables. Let $\mathbf{Z}$ be the set of all variables which are hidden in this model, which includes $\mathbf{Y}_u$, in addition to the model parameters $\mathbf{W}$, and hyper-parameters $\mathbf{A}$.

The above model can be learned, again using a variational approach (Jordan *et al.*, 1999a), in which the objective is to now to maximize the log marginal distribution

$$\log p(\mathbf{Y}_l, \mathbf{G}|\mathbf{X}) = \int_{\mathbf{Z}} q(\mathbf{Z}) \log \frac{p(\mathbf{Y}_l, \mathbf{G}, \mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} d\mathbf{Z} + KL(q||p) = \mathcal{L}(q) + KL(q||p) \tag{4.1}$$

As,

$$\mathcal{L}(q) \leq \log p(\mathbf{Y}_l, \mathbf{G}|\mathbf{X}), \tag{4.2}$$

maximizing $L(q)$ in (4.1) is equivalent to minimizing the KL divergence $KL(q||p)$. Using variational inference (Attias, 2000), (Ghahramani et Beal, 2001), let $q(\mathbf{Z})$ factorize with respect to approximate distributions for variables and approximations for distributions on parameters. The distributions for each y_j for $j \in \mathcal{U}$, and for each w_{nc} for $n = 1, 2, ..., N$ and $c = 1, 2, ..., C$, also factorize. That is, let $q(\mathbf{Z}) = q(\mathbf{W})q(\mathbf{A}) \prod_{j \in \mathcal{U}} q(y_j)$.

Expanding the terms according to the graphical model shown bove, the objective, $\mathcal{L}(q)$ decomposes into separate expectations over the unlabeled and labeled data,

$$\begin{aligned}
\mathcal{L}(q) &= \mathcal{L}_{\text{labeled}} + \mathcal{L}_{\text{unlabeled}} + \mathbb{H}[\mathbf{Z}] \qquad (4.3)\\
&= \int_{\mathbf{W}} q(\mathbf{W})q(\mathbf{A}) \log \prod_{i \in \mathcal{L}} p(y_i|\mathbf{x}_i, \mathbf{W}) d\mathbf{W} d\mathbf{A} \\
&+ \int_{\mathbf{W}} q(\mathbf{W})q(\mathbf{A}) \sum_{j \in \mathcal{U}} \sum_{y_j} \prod_j q(y_j) \log \prod_{j \in U} p(g_j|y_j)p(y_j|\mathbf{x}_j, \mathbf{W}) d\mathbf{W} d\mathbf{A} \\
&+ \int_{\mathbf{W}} q(\mathbf{W})q(\mathbf{A}) \log p(\mathbf{W}|\mathbf{A})p(\mathbf{A}) d\mathbf{W} d\mathbf{A} \\
&+ \mathbb{H}[q(\mathbf{W})q(\mathbf{A}) \prod_j q(y_j)]. \qquad (4.4)
\end{aligned}$$

The first term is identical to the strictly supervised case ; the second term encapsulates the effect of the unlabeled data. By assumption, the distributions factorize, so that the second term can be rewritten as the simpler sum of expectations

$$\begin{aligned}
&\int_{\mathbf{W}} q(\mathbf{W})q(\mathbf{A}) \sum_{j \in \mathcal{U}} \sum_{y_j} \prod_j q(y_j) \log \prod_{j \in U} p(g_j|y_j)p(y_j|\mathbf{x}_j, \mathbf{W}) d\mathbf{W} d\mathbf{A} \\
&\qquad = \sum_j \mathbb{E}_{q(\mathbf{Z})}[\log p(g_j|y_j)p(y_j|\mathbf{x}_j, \mathbf{W})]. \qquad (4.5)
\end{aligned}$$

The contribution of the unlabeled data can be interpreted as regularization, which ensures that the model prefers one in which the joint (unlabeled and labeled) model is also maximized. This is similar in spirit to the Generalized Expectation Criteria (GEC) approach of McCallum and Druck et al. (McCallum *et al.*, 2007; Druck *et al.*, 2008) in which one adds additional regularization terms to a log conditional likelihood which constrain the model by encouraging the expected values of feature functions f under the model to be close to an estimate or observation.

One can obtain similar quantities from the use of hidden variables where one maximizes expected log probabilities (e.g. $\sum_j \mathbb{E}_{q(\mathbf{Z})}[\log p(g_j|y_j)p(y_j|\mathbf{x}_j, \mathbf{W})]$) and where $\mathbf{Z}$ may encode uncertainty about parameters and/or random variables.

While the label is unobserved, the probability $p(g_j|y_j)$ can be estimated.

In this chapter, the unlabeled data is labeled with the class having the larger proportion, as the images comprising the unlabeled set are those returned from the search engine. Without loss of generality, the assumption is that the unlabeled data will be quite skewed in favor of the positive class.

Treating the distribution $p(g_j = 1|y_j)$ as Bernoulli, the straight-forward approach would be to let

$$p(g_j = 1|y_j) = \begin{cases} \mu^+ & \text{if } y_j = 1, \\ \mu^- & \text{if } y_j = -1. \end{cases}, \tag{4.6}$$

with μ^+ and μ^- constants between 0 and 1.

Before moving on to specifying the remaining quantities and deriving the updates, observe that the weak labels not only provide information, they also provide an opportunity to encode domain knowledge regarding the unlabeled data beyond just the label proportion. In what follows, however, g is simply encoded as a *discrete* random variable.

One important assumption is that classes should be separated by areas of low-density density. In the ideal case, the data is well separated by a large region of low density. Secondly, our algorithm should have low model complexity, requiring sparsity-inducing regularization.

4.1.1 Null Category

The null category is an addition to the model which has the desired property of forcing a low-density separation. Lawrence *et al.*present a null-category formulation wherein one uses an additional label for classification which takes the value zero (Lawrence et Jordan, 2005), (Lawrence *et al.*, 2005). This makes the binary classification essentially multi-class. For this reason, let $\mathbf{y}_n$ be a binary vector with 2 dimensions, with $\mathbf{y}_{n1} = 1$ if the label is positive,

and $\mathbf{y}_{n2} = 1$ if the label is negative and zero otherwise. The null category be modeled as the background class in what follows.

The restriction in (Lawrence et Jordan, 2005) and (Lawrence *et al.*, 2005) is that unlabeled data cannot take be part of the "null" category, that is, $p(Y = 0) = 0$, for unlabeled data. A maximum likelihood approach therefore heavily penalizes if unlabeled examples lie in the null category region. For a linear classifier, this has the effect of pushing decision boundaries away from the unobserved data, so that the unlabeled data must be placed in either the positive or negative region, effectively creating a region where no examples lie, providing a probabilistic margin in which no data appears.

This intuition can be incorporated into the model by appropriately parameterizing the $p(g_j = 1|y_j)$. Furthermore,the restriction $p(Y = 0) = 0$ can be relaxed to yield a "soft" margin. This still provides a penalty in high-density regions, but allows for some unlabeled examples to lie within the low density region which may be the case in many problems.

The Bernoulli distributions $p(g_j = 1|\mathbf{y}_j)$ is now modified as

$$p(g_j = 1|\mathbf{y}_j) = \begin{cases} \mu^+ & \text{if } \mathbf{y}_{j1} = 1, \\ \mu^- & \text{if } \mathbf{y}_{j2} = 1, . \\ \mu^0 = 1 - \mu^+ - \mu^- & \text{otherwise} \end{cases} \tag{4.7}$$

It is important to note that here, y and g are *discrete* labels. The effect of the null category is to attempt to bias the decision boundary *away* from unlabeled examples, that is, to be more confident that an unlabeled example is a member of one of the two labeled classes (positive or negative). This will happen when μ^0 is zero, for example, as any example lying in a null-category region will add a large penalty to the optimization. This is the original formulation of the null-category noise model. However, through experimentation, it becomes obvious in certain cases that this constraint is too strong. After experimentation, it is found that softening the constraint, *i.e.*, allowing for a small amount of probability that the label is 0 smooths the optimizations while still preferring low-density separation.

One interpretation of the model is that the null-category represents examples for which we are not sure. Because of ambiguity or problems with the weakly-labeled data gathering approach, there may be irrelevant examples – ones which do not "help" learning. Allowing the classifier to assign an example to the zero class effectively allows for the possibility that the example neither penalizes nor contributions to the objective function. This may be beneficial in some problems.

One issue is that in cases where the weak labels are always of the positive class, these values cannot be estimated from a sample, as we do not observe negative samples in a

straightforward way. To simplify μ^+ is modeled as the probability of the weak label agreeing with the true label. The probability of the two labels mis-matching is μ^-. These quantities can be estimated, as $\mu^- = 1 - \mu^+$. Leftover probability is left to the null category. Therefore, letting $\mu_j = (\mu^+)^{y_{j1}}(\mu^-)^{y_{j2}}(\mu^0)^{(1-y_{j1}-y_{j2})}$,

$$p(\mathbf{G}, \mathbf{Y}, \mathbf{W}, \mathbf{A}|\mathbf{X}) = \prod_{nc} \mathcal{N}(w_{nc}|0, \alpha_{nc})\text{Exp}(\alpha_{nc}; \gamma) \prod_{i \in L} \frac{\exp(\mathbf{y}_i^T \mathbf{W}\phi(\mathbf{x}_i))}{\sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_i))} \prod_{j \in U} \frac{\exp(\mathbf{y}_j^T \mathbf{W}\phi(\mathbf{x}_j) + \log \mu_j)}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_j))} \tag{4.8}$$

As in the supervised case, let $q(\mathbf{W}, \mathbf{A})$ factorize, with $q(\mathbf{A}) = \prod q(\alpha_{nc})$ and $q(\mathbf{W}) = \prod q(\mathbf{w}_{nc})$ and now, plugging Equation (4.8) into Equation (4.3), the objective function

$$\begin{aligned}
\mathcal{L}(q) = &\int q(\mathbf{W}) \Big(\sum_{i \in \mathcal{L}} \mathbf{y}_i^T \mathbf{W}\phi(\mathbf{x}_i) - \sum_i \log \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_i)) \Big) d\mathbf{W} \\
&+ \sum_{j \in \mathcal{U}} \sum_{\mathbf{y}_j} q(\mathbf{y}_j) \int_{\mathbf{W}} \mathbf{y}_j^T \mathbf{W}\phi(\mathbf{x}_j) - \sum_i \log(1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_j))) \Big) d\mathbf{W} \\
&+ \int q(\mathbf{W}) q(\mathbf{A}) \Big(-\frac{1}{2} \sum_{nc} \log(\alpha_{nc}) - \sum_{nc} \frac{1}{2} \alpha_{nc}^{-1} w_{nc}^2 - \frac{\gamma}{2} \sum_{nc} \alpha_{nc} \Big) d\mathbf{W} d\mathbf{A} \\
&- \sum_{j \in \mathcal{U}} \sum_{\mathbf{y}_j} q(\mathbf{y}_j) \log q(\mathbf{y}_j) - \int q(\mathbf{A}) \log q(\mathbf{A}) d\mathbf{A} - \int q(\mathbf{W}) \log q(\mathbf{W}) d\mathbf{W}.
\end{aligned} \tag{4.9}$$

As in the supervised case, $q(\mathbf{W})$ is treated as a delta function on the current estimate, $q(\mathbf{W}) = \delta(\mathbf{W} = \mathbf{W}^k)$. This allows us to avoid taking integrals over $\mathbf{W}$ analytically. Although this removes some of the elegance of the solution, q remains a variational distribution. Writing the above as expectations, we have

$$\begin{aligned}
\mathcal{L}(q) = &\sum_{i \in \mathcal{L}} \mathbf{y}_i^T \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_i) - \sum_i \log(\sum_{\mathbf{y}} \exp(\mathbf{y}^T \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_i)) \\
&+ \sum_{j \in \mathcal{U}} \log \mu_j + \langle \mathbf{y}_j^T \rangle_{q(\mathbf{y}_j)} \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_j) \qquad (4.10) \\
&- \sum_i \log(1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \langle \mathbf{W} \rangle_{q(\mathbf{W})} \phi(\mathbf{x}_j)) \\
&+ \langle \Big(-\frac{1}{2} \sum_{nc} \log(\alpha_{nc}) - \sum_{nc} \frac{1}{2} \alpha_{nc}^{-1} w_{nc}^2 - \frac{\gamma}{2} \sum_{nc} \alpha_{nc} \Big) \rangle_{q(\mathbf{A}, \mathbf{W})} + \mathbb{H}[\mathbf{Z}] \qquad (4.11)
\end{aligned}$$

The first term is simply the log likelihood of the labeled data, the second term as a prior or

regularization term. The third term is the effect of the unlabeled data, or as an additional regularization term which takes into account the prior likelihood of the classes. The final two terms penalize the entropy of the variational distributions.

Expectation The distribution $q(\mathbf{y}_j)$ can be found by noting that the bound (4.2) is tight when $\log q(\mathbf{y}_j) \propto \log p(g_j|\mathbf{y}_j)p(\mathbf{y}_j|\mathbf{x}_j, \mathbf{W})$. Dropping constant terms, the posterior distribution is then proportional to

$$q(\mathbf{y}_j) \propto \exp(\mathbf{y}_j^T \mathbf{W}\phi(\mathbf{x}_j) + \log \mu_j) \tag{4.12}$$

To simplify notation, let $\mu = (\mu^{+1}, \mu^{-1})^T$, so that the factorized distribution for each unlabeled $\mathbf{y}_j$ can be expressed as

$$q(\mathbf{y}_j) = \frac{\exp(\mathbf{y}_j^T \mathbf{W}\phi(\mathbf{x}_j) + \mathbf{y}_j^T(\log \mu_j))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_j) + \mathbf{y}_j^T(\log \mu))}. \tag{4.13}$$

This distribution is multinomial, so that the expectation is a categorical distribution where

$$\mathbb{E}[y_{jc} = 1] = \frac{\exp(\mathbf{W}_c^T \phi(\mathbf{x}_j) + \mathbf{y}_j^T(\log \mu))}{1 + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_j) + \mathbf{y}^T(\log \mu))}, \tag{4.14}$$

so that $\langle \mathbf{y}_j \rangle_{q(\mathbf{y}_j)}$ is the vector of probabilities $(q(\mathbf{y}_{j1} = 1), q(\mathbf{y}_{j2} = 1))^T$. As in the supervised case,

$$q(\tau_{nc}) \propto \tau_{nc}^{-\frac{3}{2}} \exp(-\frac{1}{2}(\gamma \tau_{nc}^{-1} + \langle w_{nc} \rangle^2_{q(w_{nc})} \tau_{nc}), \tag{4.15}$$

which is an inverse Gaussian distribution, so that again

$$\langle \tau_{nc} \rangle_{q(\tau_{nc})} = \frac{\sqrt{\gamma}}{|w_{nc}|}. \tag{4.16}$$

Again, for more details on the inverse Gaussian distribution derivation see Appendix B.

Maximization We maximize $\mathbf{W}$ again with respect to the lower bound to the joint likelihood in (4.9), where $q(\mathbf{W})$ is a delta function for which we seek a mode. With some abuse of notation, we let the expectation for observed labels $\langle y_i \rangle_{q(y_i)}$, for $i \in \mathcal{L}$, be y_i. Thus the gradient of with respect to the vector $\mathbf{W}_c$ can be written as before, with

$$\begin{aligned} \nabla_{\mathbf{W}_c} \mathcal{L}(q) = &\sum_n \langle y_{nc} \rangle_{q(y_{nc})} \phi(\mathbf{x}_n) - \sum_n \frac{\exp(\langle \mathbf{y}_{nc} \rangle_{q(y_n c)} \mathbf{W}_c \phi(\mathbf{x}_n))}{\mathbf{1}_{\{n \in \mathcal{U}\}} + \sum_{\mathbf{y}} \exp(\mathbf{y}^T \mathbf{W}\phi(\mathbf{x}_n))} \phi(\mathbf{x}_n) \\ &+ \langle \mathrm{T}_c \rangle_{q(\mathrm{T}_c)} \mathbf{w}_c, \end{aligned} \tag{4.17}$$

where $\mathbf{1}_{\{\}}$ is the indicator function and T_c is again the n by n diagonal matrix of expectations $\mathrm{diag}(\langle\tau_{nc}\rangle)_{q(\tau_{nc})}$.

The gradient is therefore a standard kernel multinomial logistic regression MAP optimization except that the indicators for the unlabeled data are replaced by expectations, in this case, $q(\mathbf{y}_j)$ We use an IRLS method with the above to maximize the parameter $\mathbf{W}$. To monitor convergence, we track the increase in the objective function (4.9), at each iteration until convergence.

To summarize, we iterate between obtaining the current estimates, with ϵ as a step-size,

$$\langle \mathbf{y}_j \rangle_{q(\mathbf{y}_j)} = (q(\mathbf{y}_{j1} = 1), q(\mathbf{y}_{j2} = 1))^T \tag{4.18}$$

$$\langle \mathbf{W}_c \rangle_{q(\mathbf{W}_c)} = \mathbf{W}_c^k - \epsilon \nabla_{\mathbf{W}_c} \mathcal{L}(q) \tag{4.19}$$

$$\langle \tau_{nc} \rangle_{q(\tau_{nc})} = \frac{\sqrt{\gamma}}{|w_{nc}|}. \tag{4.20}$$

Testing As the algorithm is not supported with weak labels at test time, a prediction is based on assuming that a test example is not unlabeled. That is, the probability $p(y = 0) = 0$. In this case, a prediction must be made, and so we use the greater of the $p(y = 1|\mathbf{x}, \mathbf{W})$ or $p(y = -1|\mathbf{x}, \mathbf{W})$. However, we note that these two probabilities are given by components of $\mathbf{W}$, namely, $\mathbf{W}_1$ and $\mathbf{W}_2$. These can be combined, $\hat{\mathbf{W}} = \mathbf{W}_1 - \mathbf{W}_2$, and $p(y = 1|\mathbf{x}) = \frac{1}{1+\exp(-\hat{\mathbf{W}}^T \phi(\mathbf{x}))}$. Instead of integrating over parameters as would be the case in a fully Bayesian procedure, we use the MAP estimate of $\mathbf{W}$ recovered from the above algorithm for computational reasons.

4.2 Experiments

In this section, we describe several experiments using this model. We first present a real-world task, attempting to increase recognition results based on the Labeled Faces in the Wild dataset using web images obtained using Google images. By thoroughly investigating this problem, we hope to give clarity to our approach. We show how unsupervised methods and simple bootstrapping are able to increase performance, but that more complex and flexible models can lead to improved performance. We then explore two simulated datasets, the often used two-moons configuration and a synthentic dataset specifically designed to mimic some properties of vision datasets to more carefully explore these ideas and our model. We then implement the model on a recognition task on the challenging Labeled Faces in the Wild dataset. To evaluate our model using this data, we deploy a set of experiments with subsets of the labeled data removed from the labeled training set. These subsets are treated as a

source of unlabeled data in order to control for the accuracy of the noise estimate. Finally we move on to a realistic task, by combining unlabeled face data obtained from Youtube videos with labeled face images from the LFW. In this last case, we also explore the effect of using a very large number of unlabeled examples on our model.

4.2.1 Google Images

A subset of the LFW dataset using the 50 subjects having the most labeled examples, yielding 2733 total labeled examples is used as the labeled data portion of the dataset. As in (Wolf *et al.*, 2008a), this set is combined with 4000 negatives drawn from the remaining subjects having as least 3 examples. This results in a subset of 6733 examples from LFW, which are used as labeled training data. The full name for each of the labeled individuals is used as search queries to download a set of images from the web using Google Image Search.

A maximum of 3000 images ranked by Google image search ranking for each subject is downloaded. The face is detected and localized in each of these images employing the OpenCV implementation of the Viola-Jones face dectector (Hannes Kruppa et Schiele, 2003), (Viola et Jones, 2001). To retain high resolution in the resulting face images, the search for faces is limited to those sized at least the maximum of 45% of the height of the video or 109×109. Each of the positive face detection is cropped and rescaled following the identical procedure as in LFW(Huang *et al.*, 2007b). A region 2.2 times the detector's bounding box width and height is selected as the crop region, to obtain the full face. If the selected area reaches outside the image boundary, the corresponding region is padded with black pixels. Finally, the expanded region is resized to 250×250. This part of the processing is designed so that the unlabeled data matches the labeled data obtained from the LFW database. This procedure resulted in 20,352 images.

The face images extracted from the videos are aligned using the funneling methodology of () for the LFW database. Following (Wolf *et al.*, 2008b), the images were then cropped to a 110×115 window around its center and converted to grayscale. An adaptive noise removal Weiner filter [1] was used for noise removal and the denoised images were normalized such that 1% of the pixels at the both the highest and lowest ends are saturated. For each of the preprocessed quality faces, features based on Local Binary Patterns (LBP features) are computed as specified by Ojala *et al.*in (Ojala *et al.*, 2002c). The four patch (FPLBP) and three patch (TPLBP) variants of LBP as described by Wolf *et al.*(Wolf *et al.*, 2008a) are concatenated into a single feature vector.

1. We use the *wiener2* filter implementation in Matlab

Top 50 LFW identities by image count	Accuracy	Std. Error
PCA (1000 dim) + NN	58.7	4.7
PCA + NN + Filtered Google	64.2	3.2
PCA -> FLDA (1000 dim) + NN	72.6	2.3
PCA -> FLDA + NN + Google***	74.2	3.0
LFW $\geq 4^*$	Accuracy	Std. Error
PCA (1000 dim) + NN	54.8	3.6
PCA + NN + Google	58.3	3.2
PCA -> FLDA (1000 dim) + NN	60.1	2.4
PCA -> FLDA + NN + Google**	67.5	4.5

Figure 4.3 Recognition accuracy for (top) the top 50 LFW identities, (bottom) All LFW identities with four or more examples, or * 610 people and 6680 images. Tests performed in a leave 2-out configuration. ** The best threshold for adding new true examples was 0.5. *** The best threshold for adding new true examples was 0.8.

In the following subsection, my colleague Fanny Puech performed and recorded several baseline nearest neighbor and bootstrapping experiments. Table 4.3 is reproduced here for comparison.

4.2.2 Baseline Experiments

A simple baseline bootstrapping experiment in which images downloaded from Google image search are selectively added to a nearest neighbour database based on their similarity to the LFW reference images for a given subject is presented here. This can be achieved by sampling the probability that two images belong to the same subject given their distance : $P(y_j = y_i|||x_j - x_i||)$. For a given distance function $d(\mathbf{x}_j, \mathbf{x}_i)$, the probability that $\mathbf{x}_j$ and $\mathbf{x}_i$ belong to the same identity is computed by sampling a large number of positive and negative examples from the LFW training set and creating distributions for positive and negative matches based on their discretized distances. Additional Google image search results are selectively added to the database based on a simple threshold on the probability that they indeed match the LFW reference images based on the average match probability across the examples in the database.

Two LFW images per identity at random to be held for testing, the rest of the images being used for both learning projections vectors and inclusion in the database. The process was repeated 20 times and we provide here the averaged accuracy.

Two distance function are evaluated. The first is based on PCA and another based on PCA followed by a non-parametric version of Fisher's LDA (Brown *et al.*, 2010). The optimization is based on maximizing $J(\mathbf{w})$ consisting of the sum of projected distances for non-match

pairs, $l_{i,j} = 0$ over the sum of projected distances for match pairs, $l_{i,j} = 1$, which can be written

$$J(\mathbf{w}) = \frac{\sum_{l_{i,j}=0}(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j))^2}{\sum_{l_{i,j}=1}(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j))^2}. \tag{4.21}$$

See (Brown *et al.*, 2010) for more details.

For each split, the features dimension is first reduced to 1000 dimensions using a PCA projection matrix learnt on the training set. Without additional images, the accuracy for the 50 LFW subjects with the most examples was 58.7 %. After we add the filtered, weakly labeled images from Google to the training-set we find that the accuracy can indeed be increased. Adding all images yields 62.1% accuracy, for a threshold of 0.5 the accuracy reaches 63.4% and for a threshold of 0.8, 64.2 %. In other words, a performance increase was possible through the use of more confident examples.

While nearest neighbour classification based on PCA projections is a widely used technique that can be made to scale well, it is a weak classification technique by modern standards. To boost performance while keeping within the simple paradigm of linear dimensionality reduction followed by nearest neighbour classification we then performed an experiment using a non-parametric variant of Fisher's LDA. Applying this projection after PCA increases our baseline accuracy to 72.6% and this bootstrapping procedure is able to yield 74.2% using a threshold of 0.8 on the average match probability.

Finally, this complete sequence of experiments for all LFW identities with four or more examples in the database is evaluated, to see how this heuristic bootstrapping procedure might scale to a problem consisting of 610 identities and 6680 corresponding images with the LFW. While performance is reduced an impressive average accuracy of 67.5% is obtianedx.

This sequence of experiments shows that a heuristic bootstrapping procedure is able to improve performance. However, a linear SVM baseline using the same amount of labeled LFW examples alone is able to yield a baseline performance of 77%. Therefore, the above approach is compared to three related methods, the first is a bootstrapped SVM where a classifier is first trained on labeled data only, and then used to classify the unlabeled examples. The results are then used to train a final classifier which is used to classify the test set. Although this method is naive, it is a commonly used procedure, and often exhibits increased performance. The second semi-supervised method is a TSVM trained transductively on the unlabeled data, and tested using the held out testing set. The TSVM was also trained in a 1-vs-all setting, with the label proportion estimated using the labeled sample of Google images. We then compare with our probabilistic method for learning with weakly labeled data.

4.2.3 Weakly Labeled Google Images

In our case, a single estimate of the weak label is not appropriate since the search rank information is available. The probability of the weak label being correct is more likely if the search ranking is higher. In order to provide a useful estimate, every identity is sampled at regular intervals along the search ranking dimension. A rank of 1 means that it appeared as the first image in the Google search. All such images with rank 1 and then every image at 20 rank intervals up to and including rank 800 (e.g. 21, 41, etc.) are sampled for each identity. Because not every rank contained images for each individual, this resulted in 952 labeled images in total across all identities, representing roughly 5% of the data.

The number of correct images divided by the number of images is regressed on the search rank. A quadratic function is used as there appeared to be an inflection point. The result of this is shown in Figure 4.4 Since the $p(g_j|\mathbf{y}_j)$ does not have to be a scalar value, the model can easily account for differing estimates reflecting confidence about the weak label. The null category estimate μ^0 was left the same for all examples.

Figure 4.4 Estimate for weak-label accuracy based on search rank.

A linear SVM is trained on features derived only from labeled images belonging to the 50 subjects, including those labeled for the noisy label accuracy estimate. Again, the test set consisted of 2 images from the Labeled Faces in the Wild dataset held out from training, resulting in 100 images. The training and testing were run 10 times across random train/test splits, resulting in an average accuracy of 79.6%. The same experiment was repeated using a Gaussian kernel resulting in a lower accuracy of 78.0%. The γ parameter in the Gaussian kernel was found by cross-validation over a grid. for these experiments, our underlying probabilistic sparse kernel classification technique trained in a completely supervised mode is able to produce comparable performance to the SVM. The SVM results are presented as the baseline accuracy, as SVMs are widely regarded as a standard state of the art classification technique.

The Bootstrap linear SVM does not show significant improvement over the linear SVM, 77.8% from 77.0%. However, in the Gaussian case, the accuracy is much improved, 73.6%. One hypothesis is that the more flexible Gaussian classifier more accurately models the input distribution, resulting in improvement when combined with unlabeled data. However this improvement still does not beat the linear SVM result on the training data alone of 77.8%. The TSVM also shows a slight but not significant improvement with a linear kernel, increasing

performance to 78.4%, but does show marked improvement in the Gaussian case, yielding 79.3%,

Using the weak label model and a null category probability μ^0, of 10% yields accuracies of 77.4% and 81.6% respectively for the linear and Gaussian kernels, respectively. These results indicate that training with a lower number of labeled examples is prone to over-fitting, requiring the use of a more restricted class of classifier, *i.e.*linear classifiers. However, in the presence of a higher number of examples in the semi-supervised case, the linear classifer underfits, and the more flexible Gaussian classifier is regularized appropriately by the unlabeled data. Table 4.2.3 summarizes these results. In this case, the linear SVM is not improved much by the addition of unlabeled data. In most cases, improvement is not statistically significant. However, although the linear classifier greatly out-performs the Gaussian in the strictly supervised setting 69.4%, the Gaussian improves greaatly after adding unlabeled data. Furthermore, the use of weak labels is able to significantly improve performance beyond the linear classifier.

4.2.4 Artificial Data

Although the effectiveness of our method on this task is shown, the following sections test some of these hypotheses more carefully. First, the hypothesis is that when labels are scarce, less flexible supervised methods, such as linear classifiers are preferred. Model complexity is desireable, however, as in the case where labels are plentiful, when unlabeled data is also available. The second hypothesis is that the null-category region allows for robustness to class-overlap.

Two Moons Dataset In order to test some intuitions about the relationship between classifier complexity and the use of unlabeled data, the semi-supervised classification technique is tested with data in the form of two moons, shown in Figure (4.5). Although a linear classifier works well when labeled data is scarce, a non-linear approach would more adequately separate the two classes. Meanwhile, using unlabeled data does not impact the linear classifier very much. However, by increasing the complexity of the classifer, the unlabeled data is able

Table 4.1 Accuracy on the LFW combined with Google Image search tested on LFW

	SVM		Bootstrap SVM		TSVM		Our Method	
Kernel	Accuracy	SE	Accuracy	SE	Accuracy	SE	Accuracy	SE
Linear	77.0	1.9	77.8	1.7	78.4	1.5	77.4	1.6
Gaussian	69.4	2.0	73.6	0.9	79.3	1.2	**81.6**	1.1

to impact the decision boundary and leads to a much more accurate result.

Previous studies on the LFW evaluation (Wolf *et al.*, 2008b) have shown that linear classifiers typically out-perform non-linear classifiers, which we also confirmed in our own experiments. Given the relatively small amount of training data, less complex, *e.g.*linear, classifiers seem to generalize better. However, when labeled training data are combined with more (noisily labeled or unlabeled) data non-linear classifiers may achieve superior performance.

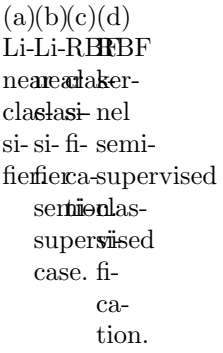


Figure 4.5 Effect of complexity and noisy label classification. The linear classifier is not improved much using a semi-supervised method. However, with a more flexible classifier, the unlabeled data is much more useful.

Figure 4.6 Data set, the green line denotes the bayes optimal classifier, training points are large circles, details in text

A Simple Example To further illustrate the model, an artificial dataset is generated by design to mimic the properties of a learning task in which one class has small intrinsic dimensionality. To produce this dataset samples are generated from a Uniform distribution in the real space $x \in [-1, 1]$. Samples representing the positive class are drawn from $y = x + e$, where e is Gaussian with standard deviation $\sigma = .1$. Samples representing the negative class are drawn from $y \sim U(-1, 1)$. The data are combined to form a 2D artificial dataset distributed according to a mixture of a Gaussian and Uniform distribution.

In order to experiment with the algorithm, an "unlabeled" set was created : all positive training points with $y > 0$ are added to the unlabeled set. Negative training points are added to the unlabeled set by sampling such that the overall accuracy of the noisy labels $(g)_{j=1,2,...,|U|}$ is equal to a specified setting.

This procedure allows for the simulation of the types of noise one might expect to see from image search results or faces extracted from videos tagged with a person's name. Namely,

unlabeled data is enriched with positive examples of the target class, but negatives are also contained within the unlabeled or weakly labeled data.

To test under realistic conditions, $g_j = 1, \forall j \in U$. In most cases, negative examples will be far easier to obtain and do not require noise parameters.

The data is divided into three equal partitions with $N = 200$ each, one for training, validation and testing. All cross-validation was performed on the validation set, which contains examples from the entire manifold and tested on the testing set, which similarly contains examples from the entire manifold. This is again, in order to mimic a realistic vision problem, in which labeled positive examples are sampled from a manifold for testing and training, but which may not have labeled examples in the training set which cover the manifold effectively.

The training data is plotted, along with the Bayes optimal decision lines superimposed in green in Figure 4.6. Thus, the data represent a classification task in which the positive class distribution has a manifold-like property, while the negative class has much less structure.

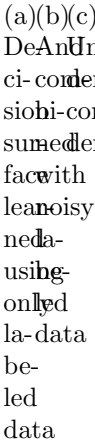


Figure 4.7 The use of noisy labels

Figure 4.7a shows the result of learning using only a portion of the data. To test the robustness and sensitivity of our noisy label classification technique to poor initialization or incorrect information about label quality, we present three experiments under varying initial estimates of the accuracy of the noisy label as shown in Figure 4.7 (b-d). Quantitative results are also shown in Table 4.2 which are averaged over 100 randomized experiments. These experiments consist of our modelling approach using : Accurate Noisy Label accuracy estimates, an Underconfident Estimate, and an Overconfident Estimate. These three configurations can be characterized by their noise estimates, μ^0 which were : .75, .5, and .99 respectively. In the experiments of Table 4.2, 100 trials are run under randomly selected sets for the unlabeled data. These unlabeled sets comprise all 43 positive examples with $y > 0$, and 14 negative

examples chosen uniformly without replacement to reach $\mu = .75$. The hyper parameter γ is chosen by cross-validation independently for the fully supervised classification task and the noisy label classification task.

Table 4.2 Results from artificial data set, showing error rates.

Error Rate Statistics for Different Methods	mean	worst	best	std	$\hat{\mu}$	std
Fully Supervised (Labeled data in Fig. 4.6)	12.00	-	-	-	-	-
Labeled Data Only (Fig. 4.7a)	33.07	35.00	30.50	7.2	-	-
Accurate Noisy Label Estimate (Fig. 4.7b)	13.50	16.50	11.00	1.5	71.8	1.6
Underconfident Estimate (Fig. 4.7c)	13.44	18.50	13.24	1.3	65.1	1.8
Overconfident Estimate (Fig. 4.7d)	14.55	18.50	11.50	1.6	99.0	.01

Table 4.2 presents the results from this data. In all three cases, the use of the noisy labels increases accuracy. While a good initial estimate leads to better overall accuracy, even choosing $\mu^0 = .99$, and $\mu^0 = .5$ lead to better solutions than using only the labeled data.

As can be seen, in Figure 4.7, the use of an overconfident estimates leads to much more confident regions, which can be interpreted as trying to fit an augmented dataset with inaccurate labelings. Meanwhile the good performance of the underconfident estimate may be the result of a reduction in variance in estimating a new label, that is, remaining ambivalent about a labeling in regions occupied by the noisy labeled examples. In general the plots indicate that the use of the most accurate estimates are indeed preferred, but remaining underconfident is also useful.

Table 4.3 Crossvalidation results on the LFW combined with Google Image search, crossvalidating for μ^0. μ^+ and μ^- are estimated from the data, leaving μ^0 as a free parameter. Again, the total probability is set to equal 1, effectively causing μ^+ and μ^- to scale but stay in proportion to each other. The results are show for Linear and Gaussian for a single held-out set. The figures in bold were used for the results shown in Table 4.2.3.

	Kernel	
μ^0	Linear	Gaussian
0	**79**	86
.10	73	**88**
.25	73	82
.5	72	81

Finally, to futher illustrate the usefulness of this additional null-category parameter, in Table 4.2.4 we show the effects of varying μ^0 within the cross-validation experiments that were used to select parameters for the LFW + Google image search results in Table 4.2.3. The validation set was a held-out test-train split. Interestingly, the linear classifier performed

best when no unlabeled images were allowed to remain unlabeled. In contrast, the cross validation for the classifier with a Gaussian kernel did not perform as well when unlabeled images were forced to take on either a positive or negative label. This may appear somewhat counter-intuitive, but since the Gaussian γ parameter is found by cross-validation, it may be that the kernel width was appropriate for the majority of examples, but under the $\mu^0 = 0$ setting a small number of examples remained ambiguous. Allowing non-zero μ^0 may result in the best compromise between the appropriate γ parameter and the null-category penalty.

4.2.5 Controlled Noise

Working with noisy labels, a natural question to ask is whether some data may be "too noisy" for use. Sources of weak labels may be easy to find, but if the data is no better than random, is it worth applying these kinds of techniques ? In fact, for most of the remaining experiments only marginally better than 50% accurate weak labels are estimated. The following experiment shows that the weak labels should be used even in marginal cases.

To investigate this issue, this section describes experiments using the Iris Plants database, first cited by Fisher (1936). The database includes 150 examples of 3 iris classes, with 4 real dimensions. The class distribution is 1/3 for each class. The very high classification rates reported using this database, as by Dasarathy (1980), in the fully labeled case makes it revealing for use in weak label noise-level experiments. The fully labeled misclassifaction rate using a linear kernel obtains 94.44% accuracy on average.

Each experiment consisted of a train-test split, where 50% of the data was used for training, 25% was used for validation, and 25% was used for testing. The training data was futher split to use 5% of each classes' examples, yielding an average of 2.5 examples per class per experiment. This is to prevent near-perfect classification rates in the labeled cases.

For each experiment, a noise-rate was chosen from 60% to 90% inclusive. In the case of 50% labels, the weak labels offers no additional information, other than to regularize, and as such was not tested. For lower than 50% accurate weak labels, the weak label algorithm defaults to basically reversing the direction of the signal (*i.e.*30% accurate weak labels, are 70% accurate weak labels for the negative class). Weak labels close to 100% are as good as labels, so the case in which weak labels are more accurate than 90% was not tested.

All model parameters (margin and regularization parameters) were chosen by evaluating on the validation set, and the experiment then tested on the remaining test set. Each of 5 kernels were tested, the linear, polynomial of order 3, polynomial of order 5, polynomial of order 7 and the RBF (Gaussian) kernel. The noise level estimates were chosen to be correct for each noise level. The experiment was repeated 25 times for each test-train split, noise-rate and kernel. The results are shown in 4.4.

	Labeled SVM		Weak Label Accuracy Rates							
			60%		70%		80%		90%	
Kernel	Accuracy	SE	Accuracy	SE	Accuracy	SE	Accuracy	SE	Accuracy	SE
Linear	75.8	1.6	64.7	3.6	64.0	2.1	62.4	4.4	67.4	1.9
Poly-3	76.9	1.5	66.3	2.2	71.9	3.4	69.6	2.8	70.7	2.7
Poly-5	75.4	1.5	70.7	2.9	74.1	3.6	77.8	3.5	79.3	3.2
Poly-7	70.7	1.8	69.1	4.2	76.1	3.6	66.6	3.8	79.9	3.6
RBF	62.3	2.4	89.7	2.5	**93.3**	1.5	91.8	1.8	93.0	1.0

Table 4.4 Accuracy rates on Iris at different weak label accuracy rates.

The results shown in 4.4 describes that in general the noise rate of the weak labels do not effect the resulting classifier as much as accurately modeling the noise. In this case, because the weak label noise was directly manipulated, these exact values were used in training the classifiers. Although in general more accurate weak labels do yield better classifier accuracy, the results do not dramatically differ, and indeed are not statistically significant within a kernel type.

However, an interesting artifact of this experiment is the effect of classifier complexity. The effectiveness of the weak labels increase as the complexity increases, while the opposite is true for the labeled-only case, which appears to overfit. It should be noted that the p-value for the RBF kernel weak label classifier versus the linear SVM is less than .0001 ;

4.2.6 Labeled Faces in the Wild - Controlled Noise Experiments

Although the artificial data in the previous section mimics properties of vision data, to test the accuracy of our method on real labeled data, the model is further tested under a known weak label accuracy level. The dataset created in the previous section for our Google Images experiment is again used. The algorithm is tested using varying amounts of labeled examples by holding out a certain percentage of each subject's images, and creating a synthetic unlabeled set with 75% accurate weak labels from the labeled portion of the data. This allows for testing using a known weak-label accuracy.

Table 4.5 Accuracy using different proportions of labeled and unlabeled data using a known weak label accuracy parameter. The held out column presents the percentage of data used as unlabeled data. For our method, γ and μ^0 were set to 1 and .50, respectively. To set μ^+ and μ^- are determined by the value of μ^0 by setting $\mu^+ = .75(1 - \mu^0)$, and $\mu^- = .25(1 - \mu^0)$, the proportion of the remaining probability. Here we note that these values were found during crossvalidation, and that the classification rate during cross-validation using $\mu^0 = 0$, which would be the most similar to using the null-category noise model was, on average, 14.75% lower.

	SVM		TSVM		Our Method + Noisy Labels	
Held out	Accuracy	SE	Accuracy	SE	Accuracy	SE
All but 1	23.6	1.4	57.9	1.1	**60.3**	1.2
0.9000	44.9	1.2	58.4	1.7	**61.0**	0.4
0.7500	60.4	1.3	**65.7**	1.9	65.1	0.4
0.5000	70.9	1.3	70.6	0.9	**75.4**	0.1

The data in Table (4.2.6) shows the results of using a linear kernel with the LBP features as input for each held-out regime, with parameters determined by cross-validation. The results are also compared with the TSVM. The results indicate that a significant increase in accuracy is possible using unlabeled data, especially in the case where only a very small number of positive labels are available. The additional improvement decreases as the ratio of labeled training examples to unlabeled examples increases. This is true for both the TSVM and our method. The results for the TSVM and our method are quite similar, however, our method shows the greatest improvement when more labeled data is available.

4.2.7 Youtube Video

The subset of identities obtained above and the corresponding names for these individuals are then used to download a set of videos from YouTube, again using their full names in quotations as queries. A maximum of thirty videos ranked by YouTube's search is downloaded for each subject. To avoid returning near duplicates, faces are collected from only key frames using () and MEncoder tools. Again, the OpenCV implementation of the Viola-Jones face dectector and preprocessing steps in the previously described experiments are used to keep the weakly labeled data as similar to the LFW images as possible.

After processing 1277 downloaded videos, a total number of $42,255$ faces were extracted. False negative face detections are filtered out by running a eye-pair detector provided by OpenCV on the resulting images (Hannes Kruppa et Schiele, 2003), resulting in $25,726$ face images.

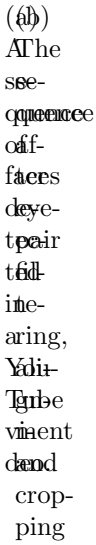


Figure 4.8 The pipeline output for one of Winona Ryder's videos

The face images extracted from the videos are aligned using the funneling methodology of () for the LFW database. Following (Wolf *et al.*, 2008b), the images were then cropped to a 110×115 window around its center and converted to grayscale. An adaptive noise removal filter (*wiener2*) and the denoised images were normalized such that 1% of the pixels at the both the highest and lowest ends are saturated. For each of the preprocessed quality faces, the LBP, (Ojala *et al.*, 2002c), FPLBP and TPLBP as described in (Wolf *et al.*, 2008a) are concatenated as the features. The pipeline is described visually in Figure 4.8.

A small amount of labels for each video to estimate the weak-label parameters μ, labeling 4,473 random images of the available 20,765 facial images by random sampling – about 90 for each subject. This provided 2,369 additional positive examples for 50 subjects. Since each subject's estimate varied widely, these estimates were regressed to the global mean of 53%, which effectively provided a prior to the binomial distribution. These estimates were combined with a relatively large margin size (μ^0 of 50% found by cross-validation on test/train split), to yield a multinomial distribution. A linear SVM was trained on features of the labeled YouTube face images and evaluated on testing samples of 100 images comprised of two images for each subject from a held-out set. In this case, 2 labeled images for each of the YouTube images belonging to the identity is held-out from training. Both of these images are taken from the same YouTube video, with the other images from the video also held-out from training.

The training and testing were again run 10 times across different train/test splits, resulting in an accuracy of 81.6%. The same experiment was repeated using a Gaussian kernel resulting in a lower accuracy of 78.0%. The γ parameter in the Gaussian kernel was found by cross-

validation over a grid. When combined with LFW data, using a null category probability μ^0, of 75% yielded accuracies of 75.6% and 85.8% respectively for the linear and Gaussian kernels. In the semi-supervised case, the linear classifer under-fits, and the more flexible Gaussian classifier is regularized appropriately by the unlabeled data. Table (4.6) summarizes these results.

Table 4.6 Accuracy on the YouTube faces.

	SVM		Our Method	
Kernel	Accuracy	SE	Accuracy	SE
Linear	81.6	1.9	77.0	1.5
Gaussian	78.0	0.9	**85.8**	1.2

Adding the Youtube facial images to the LFW dataset yielded increased performance using the model based on the larger number of positive examples in the unlabeled set. The results of augmenting LFW training with our Youtube data are presented in Figure 4.2.7. Interestingly, here again a more flexible classifier increases performance. The baseline experiment, as described previously is a linear classifier trained using all but 2 of the available labeled LFW images, which were used for testing. These were combined with the labeled examples from the YouTube data. Both a linear and Gaussian SVM were trained using only the labeled data, and then our method was used by adding the remaining unlabeled Youtube images. Again, this set of experiments was repeated 10 times over differing train/test splits. Similar to the Youtube-only experiments, it is apparent that the more flexible Gaussian kernel is preferable in the semi-supervised case. The linear kernel case indicates serious underfitting.

Table 4.7 Accuracy on the LFW using the LFW augmented with Youtube faces as training data.

	SVM		Our Method + Noisy Labels	
Kernel	Accuracy	SE	Accuracy	SE
Linear	78.1	0.7	64.6	1.6
Gaussian	76.3	3.5	**81.8**	1.3

x The above experiment on a final task, to attempt to combine static images with unlabeled video images in order to better classify faces found in the video images, *i.e.*train using the same dataset but test on video images. This experiment is highly relevant to the practical application of tagging faces in video. Again, the unlabeled data helps to create a better classifier for the video, as shown in Table 4.2.7. In this case, training on images drawn

from videos present in the test set was prevented by removing those images from the test sampling procedure.

Table 4.8 Accuracy on the YouTube faces using the LFW augmented with Youtube faces as training data.

	SVM		Our Method + Noisy Labels	
Kernel	Accuracy	SE	Accuracy	SE
Linear	82.3	0.4	71.0	1.8
Gaussian	77.5	1.0	**86.1**	0.5

4.3 Discussion

Transferring information across the modalities of static images and video can be quite challenging. Facial images drawn from the LFW dataset are derived from static news images which contain faces that are usually centered and well posed in the photo. In contrast, the facial images drawn from Youtube exhibit quite a large amount of variability due to the continuous nature of video, differences in environments and compression artifacts. A direction for future work would include accounting for these types of differences as we believe one of the main factors limiting the performance of our approach for this problem is domain differences.

However, despite the challenges of transferring information across domains, this approach is able to boost performance in all cases. For the last experiment, in which the feasibility of improving classifiers trained on static images to improve face tagging in video, performance increases from 82% accuracy to 86% accuracy. These results are both statistically significant and in terms of the types of improvements one sees on the LFW, a 4% increase is not negligible. The use of a probabilistic null-category model with soft constraints and a novel prior produces better results than using the labeled data alone.

CHAPTER 5

Identity Normalization For Facial Expression Recognition

The previous chapter focused on face recognition, more specifically, improving facial recognition performance using weakly labeled data. This chapter deals with improving facial expression recognition by leveraging information that might be considered to be auxiliary to the task. In particular, this chapter deals with probabilistic modeling identity information to help with expression recognition. Portions of this chapter have been submitted to the International Journal of Computer Vision.

One of the primary sources of variation in facial images is identity. Although this is an obvious statement, many approaches to vision tasks other than facial recognition do not directly account for the interaction between identity-related variation and other sources. However, many facial image datasets are subdivided by subject identity. This provides additional information. This chapter deals with the natural question of how to effectively use identity information in order to improve tasks other than identity recognition.

Recently there has been work on facial recognition in which identity is separated from other sources of variation in 2D image data in a fully probabilistic way (Prince *et al.*, 2011). In this model, the factors are assumed to be additive and independent. This procedure can be interpreted as a probabilistic version of Canonical Correlation Analysis (CCA) presented by Bach (Bach et Jordan, 2005), or as a standard factor analysis with a particular structure in the factors.

The approach allows for the full use of expression datasets that are not uniform across all subjects. In this chapter, we investigate and extend the use of this probabilistic approach for expression and facial animation tasks. This includes performance-driven animation, emotion recognition, and key-point tracking. Because these tasks require representations which should generalize across identities, we show how appropriately factorizing the input representations can yield improved results. In many cases we use the learned representations as input to discriminative classification methods.

In our experimental work we apply this learning technique to a wide variety of different input types including : raw pixels, key-points and the pixels of warped images. We evaluate our approach by predicting : standard emotion labels, facial action units, and 'bone' positions or animation sliders which are widely used in computer animation. We also improve the prediction performance of active appearance models (AAMs) on unseen identities. Our evaluations are summarized in Table 5.1. Below we discuss these applications in more detail

and review relevant previous work.

5.1 Literature Review

The literature of expression recognition was covered in detail in previous chapters. There are many approaches to expression recognition which directly attempt to account for identity. However, many of these approaches are either complex or do not make full use of the data. In this chapter, the approach to identity normalization for expression recognition is to simplify the generative model used in multi-linear analysis but to preserve the main idea – that of separating identity factors of variation from expression data – in a probabilistic way. Moreover, like multi-linear analyses, the method is useful for other tasks. In this chapter, we show how to use the method for performance-driven animation and key-point localization.

5.1.1 Performance-driven Facial animation

Many marker-less facial expression analysis methods rely either on tracking points using optical flow (Essa *et al.*, 1996) or fitting Active Appearance Models (Lanitis *et al.*, 1997). Morphable models in (Blanz et Vetter, 1999), (Pighin *et al.*, 1999), (Sibbing *et al.*, 2009) have also been investigated. More recently, 3D data and reconstruction is used to fit directly to the performer (Wang *et al.*, 2004a), (Zhang *et al.*, 2007). These methods, however, often require additional data, for example, multi-view stereo (Yin *et al.*, 2008), () or structured light (Chang *et al.*, 2005), (Wang *et al.*, 2004a). Sandbach *et al.*provide a thorough review of 3D expression recognition techniques in (Sandbach *et al.*, 2012). In the end, these methods usually work by providing dense correspondences which then require a re-targeting step.

However, a simpler and often used approach in industry does not rely on markers and simply uses the input video of a facial performance. The idea is to use a direct 2D to 3D mapping based on regressing image features (Goudeaux *et al.*, 2001) to 3D model parameters. This method works well but is insufficiently automatic. Each video is mapped to 3D model parameters, possibly with interpolation between frames. Key-point based representations often require both data and training time that compares unfavorably to this simpler approach given the re-targeting step. The primary benefit of key-point based representations appears to be a degree of natural identity-invariance. We provide experiments on performance driven animation, in which we predict bone positions using the well known Japanese Female Facial Expression (JAFFE) Database (Lyons *et al.*, 1998) as input as well as an experiment using professional helmet camera based video used for high quality, real world animation animations.

5.1.2 Key-point Localization

A widely used approach to key-point detection relies on point distribution models, including Active Appearance Model's (AAM) (Edwards *et al.*, 1998), (Cootes *et al.*, 1995a) and Locally Constrained Models (LCM) (Saragih *et al.*, 2010). These usually suffer from a degree of identity-dependence. That is, a model trained on a sample of subjects does not necessarily perform well on an unseen subject.

n the AAM, a set of key-points are predefined for an object class – typically chosen as candidate points for controlling the curvature of a fitted spline around object boundaries. A "shape" can be described as a particular positioning of these landmarks. Each shape is therefore a vector of control point coordinates. Each training example is aligned to a common orientation, using an energy minimization procedure based on a Mehalanobis distance of each training shape to the mean shape, after which each landmark is treated independently.

Using PCA on the transformed images recovers a sub-space representation of shape. The "Active" part of the shape model refers to how models are fitted to test examples – iteratively updating the position of each landmark starting from some initialization by alternately fitting local updates to "reasonable" regions of the shape space. The addition of intensity information, (similar to the procedure of the eigen-face) via vector concatenation gives a natural combined model which can be used to generate the original image. That is, first computing a shape space and an eigen-space for pixel intensity. Then using a fitted shape model to deform the fitted eigen-face.

The EBGM is a similar procedure in which the points are accompanied with local features in the form of Gabor Jet Descriptors (GJD). Graph matching is then performed not only by landmark position, but also by comparison of the local information. To fit a graph to a test example iteratively searches over the Gabor space locally while constrained by a global graph term. The global graph term can be thought of as an energy term penalizing the "elasticity" of the graph. These and other similar graph based methods can deal much better with sources of variance such as pose, scale and occlusions, by deriving a shape based representation which attempt to explain facial identity through graph matching.

The AAM, in particular, performs much better when samples of an individual are used for both training and testing. In this chapter, the problem of when no additional information about a new subject is available, as is the case in many common scenarios, is investigated. As such, the method is evaluated using the AAM as a test algorithm, because it performs poorly in this setting but performs well otherwise.

View-based approaches (Pentland *et al.*, 1994b), and multi-stage solutions (Tistarelli et Nixon, 2009), (Liao *et al.*, 2004) address this issue by using multiple subspaces for each identity. Gaussian mixture models, (Frey et Nebojsa, 1999) indirectly deal with identity

variation by learning clusters of training data. Gross *et al.*described reduced fitting robustness of AAMs on unseen subjects (Gross *et al.*, 2005), suggesting simultaneous appearance and shape fitting improve generalization.

5.2 Model

The main assumption is that there are two primary sources of variation in the dataset. A graphical model of this approach is shown in Figure 5.1. For this model, we take as inputs observed data $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,2,\ldots,N}$ composed of vectors of features that can be divided into sets by identity. In our case these can be pixels, vectors containing key-point locations or a concatenation of both. For a collection of I identities with J_i images per identity, we use the notation $\mathbf{x}_{ij}$ to denote the j^{th} image of the i^{th} identity in the dataset.

Figure 5.1 Graphical model of facial data generation. $\mathbf{x}_{ij}$ is generated from $p(\mathbf{x}_{ij}|\mathbf{v}_{ij}, \mathbf{w}_i)$, after sampling $\mathbf{w}_i$ from an identity and $\mathbf{v}_{ij}$ from an expression-related distribution respectively.

Each $\mathbf{x}_{ij}$ is generated by sampling $\mathbf{w}_i$ and $\mathbf{v}_{ij}$ from Gaussian distributions corresponding to identity i and expression i_j distributions $p(\mathbf{w}_i)$ and $p(\mathbf{v}_{ij})$, and then combining these by sampling an image $\mathbf{x}_{ij}$ according to $p(\mathbf{x}_{ij}|\mathbf{w}_i, \mathbf{v}_{ij})$. We use zero-mean independent Gaussian distributions for $p(\mathbf{w}_i)$ and $p(\mathbf{v}_{ij})$,

$$p(\mathbf{w}_i) = \mathcal{N}(\mathbf{w}_i; \mathbf{0}, \lambda\mathbf{I}) \tag{5.1}$$

$$p(\mathbf{v}_{ij}) = \mathcal{N}(\mathbf{v}_{ij}; \mathbf{0}, \rho\mathbf{I}). \tag{5.2}$$

Next, $\mathbf{x}_{ij}$ is then sampled from a multivariate Gaussian conditional distribution parameterized by the mean μ, matrices $\mathbf{F}, \mathbf{G}$ and diagonal covariance $\mathbf{\Sigma}$.

$$p(\mathbf{x}_{ij}|\mathbf{w}_i, \mathbf{v}_{ij}) = \mathcal{N}(\mathbf{x}_{ij}; \mu + \mathbf{F}\mathbf{w}_i + \mathbf{G}\mathbf{v}_{ij}, \mathbf{\Sigma}) \tag{5.3}$$

This corresponds to a conditional distribution with variables $\mathbf{w}_i$ which are common for all images of a unique identity and $\mathbf{v}_{ij}$ which are allowed to vary across a particular identity. However, the loadings $\mathbf{F}$ and $\mathbf{G}$ themselves are shared across all identities, so that $\mathbf{F}$ corresponds to loadings that are associated with identity and $\mathbf{G}$ those of expression. The joint probability can be written as

$$p(\mathbf{X}, \mathbf{w}, \mathbf{v}|\mathbf{F}, \mathbf{G}, \lambda, \rho) = \prod_{i}^{I} \mathcal{N}(\mathbf{w}_i; \mathbf{0}, \lambda\mathbf{I}) \tag{5.4}$$
$$\prod_{j}^{J_i} \mathcal{N}(\mathbf{x}_{ij}; \mu + \mathbf{F}\mathbf{w_i} + \mathbf{G}\mathbf{v_{ij}}, \mathbf{\Sigma}) \mathcal{N}(\mathbf{v_{ij}}; \mathbf{0}, \rho\mathbf{I}).$$

We can then use Expectation Maximization (EM) to learn the parameters of the model $\theta = \{\mathbf{F}, \mathbf{G}, \Sigma\}$.

5.2.1 Learning

As with any EM algorithm optimization, the goal is to maximize the joint distribution by alternatively maximizing and taking expectations

$$\max_{\theta} \mathbb{E}[\log p(\mathbf{X}, \mathbf{w}, \mathbf{v}|\theta)], \tag{5.5}$$

where the expectation is with respect to the posterior conditional distribution $p(\mathbf{w}, \mathbf{v}|\mathbf{X}, \theta^{\text{old}})$. As the distributions are both Gaussian, the resulting distribution is Gaussian as well as shown in (Prince *et al.*, 2011).

The observed variables $\mathbf{x}_{iv}$, can be written as a single feature vector for each identity i, as

$$\mathbf{x}_i = (\mathbf{x}_{i1}^T, \mathbf{x}_{i2}^T, \ldots, \mathbf{x}_{iJ_i}^T)^T \tag{5.6}$$

Similarly the factors can be combined into a single loading matrix

$$\mathbf{A} = \begin{pmatrix} \mathbf{F} & \mathbf{G} & 0 & ... & 0 \\ \mathbf{F} & 0 & \mathbf{G} & ... & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{F} & 0 & 0 & ... & \mathbf{G} \end{pmatrix}, \tag{5.7}$$

with factors

$$\mathbf{d}_i = (\mathbf{w}_i^T, \mathbf{v}_{i1}^T, \mathbf{v}_{i2}^T, \ldots, \mathbf{v}_{iJ_i}^T)^T. \tag{5.8}$$

Given this construction, the probability for identity i can be rewritten as a Gaussian $p(\mathbf{x}_i|\mathbf{d}_i) = \mathcal{N}(\mathbf{x}_i; (\mu + \mathbf{A}\mathbf{d}_i), \Psi)$, with Ψ constructed as a diagonal matrix with Σ along the diagonal repeated J times. Since $\mathbf{d}$ is composed of two vectors with zero-mean Gaussian distributions, it is also distributed as a zero mean Gaussian. The posterior probability of $\mathbf{d}_i$ is also Gaussian,

with diagonal variance Γ. Γ is zeros except for λ and ρ along the diagonal in the form λ for rows $1, 2, ..., K$, where $\mathbf{w} \in R^K$ and ρ for rows $K+1, K+2, ..., J_i * (K+L)$, for $\mathbf{v}_{ij}$ in R^L. Given this construction, the posterior distribution is Gaussian with moments

$$\mathbb{E}[\mathbf{d}_i] = (\mathbf{A}^T \Psi \mathbf{A} + \mathbf{I}\sigma)^{-1} \mathbf{A} \Psi (\mathbf{x}_i - \mu) \tag{5.9}$$

$$\mathbb{E}[\mathbf{d}_i \mathbf{d}_i^T] = (\mathbf{A}^T \Psi \mathbf{A} + \mathbf{I}\sigma) - \mathbb{E}[\mathbf{d}_i]\mathbb{E}[\mathbf{d}_i^T]. \tag{5.10}$$

The joint distribution can be maximized for each identity separately. For a single identity i, we can also re-write the joint in terms of $\mathbf{d}_{ij}$.

$$\max_{\mathbf{B}} \sum_j (\mathbf{x}_{ij} - \mathbf{B}\mathbf{c}_{ij})^T \Sigma^{-1} (\mathbf{x}_{ij} - \mathbf{B}\mathbf{c}_{ij}) - \frac{1}{2} \log |\Sigma^{-1}|,$$

where $\mathbf{B}\mathbf{c}_{ij} = \mathbf{F}\mathbf{w}_i + \mathbf{G}\mathbf{v}_{ij}$, splitting the components of $\mathbf{c}_i$ appropriately. Maximizing with respect to $\mathbf{F}, \mathbf{G}$ and Σ, we have the following updates

$$\begin{aligned}
\mathbf{F} &= \big(\sum_{ij} \mathbf{x}_{ij} \mathbb{E}[\mathbf{w}_i]^T \big) \big(\sum_{ij} \mathbb{E}[\mathbf{w}_i \mathbf{w}_i^T] \big)^{-1}, \\
\mathbf{G} &= \big(\sum_{ij} \mathbf{x}_{ij} \mathbb{E}[\mathbf{v}_{ij}]^T \big) \big(\sum_{ij} \mathbb{E}[\mathbf{v}_{ij} \mathbf{v}_{ij}^T] \big)^{-1}, \\
\Sigma &= \frac{1}{N} \mathrm{diag}\big(\sum_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^T - (\mathbf{F}\mathbb{E}[\mathbf{w}_i] - \mathbf{G}\mathbb{E}[\mathbf{v}_{ij}]) \mathbf{x}_{ij}^T \big).
\end{aligned}$$

As for inference at test time the procedure for determining the optimal $\mathbf{w}$ and $\mathbf{v}_j$ vectors is straight-forward, using the posterior distribution.

$$p(\mathbf{d}_i | \mathbf{x}_i) = \mathcal{N}(\mathbf{d}_i; (\mathbf{A}^T \Psi \mathbf{A} + \mathbf{I}\sigma)^{-1} \mathbf{A} \Psi (\mathbf{x}_i - \mu), \mathbf{A}^T \Psi \mathbf{A} + \mathbf{I}\sigma) \tag{5.11}$$

The conditional distributions of $p(\mathbf{w} | \mathbf{v}_j, \mathbf{x}_i)$ and $p(\mathbf{v}_j | \mathbf{w}, \mathbf{x}_i)$ can be derived easily as well.

5.3 Experiments

Table 5.1 Summary of experiments described in this paper, numbers of the section describing each experiment are given in parenthesis.

JAFFE	
Emotion Recognition	(5.3.1)
predicts :	Emotion labels
using :	Images
Animation Control	(5.3.2)
predicts :	Bone position parameters
using :	Images
Extended Cohn-Kanade	
Facial Action Unit (AU) Detection	(5.3.1)
predicts :	AU labels
using :	Point locations
	Shape-normalized images
	Combined point locations and shape-normalized images
Emotion Recognition	(5.3.1)
predicts :	Emotion labels
using :	Point locations
	Shape-normalized images
	Combined point locations and shape-normalized images
Key-Point Localization	(5.4.1)
predicts :	Key-point locations
using :	Images
Animation Control Studio Data	
Animation Control	(5.3.2)
predicts :	Bone position parameters
using :	Shape-normalized images

In the following sections, we present results of applying the model to emotion recognition and performance-driven animation using multiple datasets. We first present emotion recognition results, including facial action unit classification. Then we move on to performance-driven animation experiments. We also evaluate the task of key-point detection in Section 4. We show that the factorization model described above yields almost automatic improvement in performance across this wide variety of applications. using standard approaches. A summary

of these experiments is shown in Table 5.1.

5.3.1 Emotion Recognition

JAFFE

Figure 5.2 The JAFFE dataset contains 213 labeled examples for 10 subjects. Images for a single test subject, left out from training, shown here with predicted labels from our method. The data is shown in two sets of rows. The first row in each set is the original input data, the second a rendering of the mesh with corresponding bone positions predicted by the model.

We first run a set of experiments on the constrained JAFFE dataset (Lyons *et al.*, 1998). The JAFFE dataset contains ten identities with varying expression across seven emotions containing 213 images in all. The frontal facial images are roughly aligned but we use a funneling algorithm (Huang *et al.*, 2007a) after face detection to correct small pose variation.

We then use an ellipse, manually specified, to mask background variation and reduce dimensionality, and then divide the face into three rectangular regions roughly corresponding to the mouth, eyes and ears. Unlike previous work, however, (Zhou et Lin, 2005), we do not manually locate facial landmarks. Then we run the algorithm for each region individually, learning a composite space in which facial expressions are treated as a linear combinations. The expression space weights for each region are learned separately and then concatenated to form a single input vector consisting of the weights $\mathbf{v}_{ij}$.

We then predict seven emotional states – anger, disgust, fear, happiness, neutral, sadness

and surprise – using the learned representation, $\mathbf{v}_{ij}$. We train an SVM RBF classifier for each of seven emotional states on the in-sample subject images, using a single image from each of the emotions of the left-out subject as a validation set to learn the slack and kernel bandwidth parameters. We treat the classification at test time as a one-against-all prediction. We compare our results against PCA performed on the same input data as our model, using 30 and 100 dimensions. The final rates are shown in Table 5.2 as the average accuracy across identities.

The overall prediction accuracy rate for the emotion prediction for an unseen identity is 72.17% using a 30 dimensional expression space. A recent result on the JAFFE set by Cheng *et al.* (Cheng *et al.*, 2010), in experiments with left-out subjects, obtained a mean accuracy of 55.24% using Gaussian process classification. Because their pre-processing is not identical and they do not report standard errors, we do not include it in Table 5.2.

Extended Cohn-Kanade

Certainly, the JAFFE data exhibits far less variation than is usually present in data. In this section we present more detailed experiments using the Extended Cohn Kanade (CK+) database (Lucey *et al.*, 2010) for emotion recognition and facial action unit tasks. The CK+ dataset consists of 593 image sequences from 123 subjects ranging in age from 18 to 50, 69% of whom are female and 13% of whom are black. The images are frontal images of posed subjects taken from video sequences. Each sequence contains a subject posing a single facial expression starting from a neutral position. The sequence consists of sampled frames from video in which the final posed position is labeled with FACS action units. In addition, emotion labels, consisting of the expressions "Anger," "Disgust," "Fear," "Happiness," "Sadness," "Surprise" and "Contempt", are provided for 327 of the 593 sequences.

Table 5.2 Accuracy for JAFFE emotion recognition in percentage and Mean Squared Error for bone position recovery experiments for JAFFE and Studio Motion Capture data, calculated per bone position, which lie in $[-1, 1]$.

	JAFFE				Studio Data	
	Emotion Recognition		Bone Position Recovery		Bone Position Recovery	
	Accuracy	SE	MSE	SE	MSE	SE
No Factor Analysis	53.13%	3.39%	1.7526	0.2702	1.5809	0.2300
PCA 100 dimensions	57.08%	6.57%	1.7526	0.2702	0.0786	0.0120
PCA 30 dimensions	56.13%	5.64%	0.1223	0.0121	0.1007	0.0123
Our Method	**72.71%**	1.83%	**0.0851**	0.0077	**0.0231**	0.0021

Facial Action Unit Detection The CK+ database contains 593 labeled sequences, however only the final image as well as the initial neutral image may be used for traditional classification. Lucey *et al.*(Lucey *et al.*, 2010) describe their baseline approach based on linear classification of key-points and warped images obtained by Active Appearance Models. The resulting landmark data is also provided in this dataset. To compare our method with the baseline, we recreate their approach. First, we use a Procrustes analysis using affine transformation of the landmark positions to determine a mean shape and register the point locations to the mean by estimating the least square best 2D transformation. We then run PCA on the difference between the Procrustes aligned points and the original points to determine the similarity components. We add these to the AAM shape parameters to model rigid motion. Finally, a piece-wise affine warp is applied to normalize the shape of each facial image to the base shape recovered from the Procrustes analysis to obtain warped images.

Using this approach, we generate two feature sets. The first is point locations after Procrustes analysis. The second are the shape-normalized images, which are converted to vectors. We use leave-one-subject-out cross-validation using linear SVM's for each of the 17 AU's on the vectors of landmark locations and vector of shape-normalized images, and record the AUC score for each left-out subject and AU pair. An estimate for the AUC error is calculated as well, defined as $\sqrt{\frac{A(1-A)}{\min(N_p,N_n)}}$, where A denotes the AUC score and N_p, N_n are the number of positive and negative examples respectively. To combine feature sets, Lucey *et al.*run logistic regression on the scores of SVM's built from the two feature-sets independently. We also recreate this step.

For our identity-normalization experiments, we use the point-locations and shape-normalized images as inputs to our factor analysis. In this case, we use 100 and 30 dimensions for the identity and expression parameter vectors respectively. In order to avoid over-fitting, we increase the training data size for this unsupervised step from 1186 to 2588 by ensuring that each identity has at least 20 images, by sampling uniformly from intermediate frames. One subject is used for validation (subject 1), but we show results including this subject as it represents the leave-2-subject out cross-validation results used in (Lucey *et al.*, 2010).

One important consideration is that no testing subject images are used during training our models, including during the factor analysis, in order to maintain fairness.

3blue!15white

Table 5.3 CK+ : AUC Results and estimated standard errors of the AU experiment

CK+ Facial Action Unit Recognition

		Baseline Lucey *et al.* (2010)			Identity Normalized		
AU	N	SPTS	CAPP	SPTS+CAPP	SPTS	CAPP	SPTS+CAPP
1	173	94.1 ± 1.8	91.3 ± 2.1	96.9 ± 1.3	97.7 ± 2.8	96.33 ± 3.2	**99.1 ± 1.2**
2	116	97.1 ± 1.5	95.6 ± 1.9	97.9 ± 1.3	97.2 ± 2.4	97.82 ± 1.0	**98.5 ± 0.9**
4	191	85.9 ± 2.5	83.5 ± 2.7	91.0 ± 2.1	89.6 ± 9.3	91.72 ± 7.7	**94.3 ± 5.6**
5	102	95.1 ± 2.1	96.6 ± 1.8	97.8 ± 1.5	96.1 ± 2.1	97.52 ± 2.6	**98.0 ± 1.6**
6	122	91.7 ± 2.5	94.0 ± 2.2	95.8 ± 1.8	96.8 ± 3.3	95.37 ± 4.1	**97.3 ± 2.8**
7	119	78.4 ± 3.8	85.8 ± 3.2	89.2 ± 2.9	89.7 ± 7.4	92.95 ± 7.1	**94.7 ± 5.9**
9	74	97.7 ± 1.7	99.3 ± 1.0	99.6 ± 0.7	99.2 ± 0.8	**99.71 ± 0.4**	98.2 ± 0.2
11	33	72.5 ± 7.8	82.0 ± 6.7	85.2 ± 6.2	81.1 ± 4.1	83.15 ± 4.2	**91.6 ± 3.6**
12	111	91.0 ± 2.7	96.0 ± 1.9	96.3 ± 1.8	96.9 ± 2.2	**97.68 ± 2.2**	97.0 ± 2.0
15	89	79.6 ± 4.3	88.3 ± 3.4	89.9 ± 3.2	90.2 ± 4.2	96.42 ± 2.0	**96.8 ± 2.5**
17	196	84.4 ± 2.6	90.4 ± 2.1	93.3 ± 1.8	92.1 ± 6.7	95.48 ± 4.2	**96.3 ± 3.6**
20	77	91.0 ± 3.3	93.0 ± 2.9	94.7 ± 2.6	95.9 ± 3.4	94.62 ± 2.4	**96.7 ± 1.7**
23	59	91.1 ± 3.7	87.6 ± 4.3	92.2 ± 3.5	91.7 ± 3.7	93.77 ± 2.9	**96.5 ± 2.8**
24	57	83.3 ± 4.9	90.4 ± 3.9	91.3 ± 3.7	88.9 ± 4.5	92.91 ± 3.4	**94.2 ± 3.4**
25	287	97.1 ± 1.0	94.0 ± 1.4	97.5 ± 0.9	97.6 ± 2.6	96.83 ± 3.6	**98.1 ± 1.9**
26	48	75.0 ± 6.3	77.6 ± 6.0	80.3 ± 5.7	77.9 ± 8.1	83.61 ± 7.4	**86.6 ± 6.8**
27	81	99.7 ± 0.7	98.6 ± 1.3	99.8 ± 0.5	99.9 ± 0.2	99.67 ± 0.5	**99.9 ± 0.2**
AVG		90.0 ± 2.5	91.4 ± 2.4	94.5 ± 2.0	92.8 ± 4.0	94.45 ± 3.5	**96.1 ± 2.7**

The results are shown in Table 5.3, which shows an increase in average AUC for each of the feature-sets individually. In a majority of AU classes, the AUC score increased, while there are no large reductions in AUC after identity normalization. For both the SPTS and CAPP scores, identity normalization showed significant increases in performance for AU-7, Lid-Tightener and AU-15, Lip Corner Depresser. AU-7 is associated with an eye-narrowing or squinting action, and AU-15. We show examples from this dataset of these two AU's to show that resulting normalization leads to representations that can be seen as roughly identical.

In both cases, the expression is well reproduced in the identity-normalization procedure. The increase in performance from using both feature-sets is apparent again, with greater than .86 AUC for every AU and most well over 90%. From our comparison, it is clear that the simple step of identity normalization improves performance on average. It appears to work especially well for those AU's that are easy, in some sense, to capture between identities.

Emotion Detection The baseline approach can also be used for emotion recognition. Again, we recreate their approach and use the identity-normalization step as an unsupervised learning procedure. Because of this, we can use more training examples than those that are labeled, making this procedure semi-supervised. Again, we use the 2588 images used in the AU experiments. Each of the 327 labeled examples along with the neutral examples are then

projected into the space learned in the previous step and used as input for training the linear SVM's. In this case, each SVM learns a one-vs-all binary classifier for the emotion of interest, using the rest as negative examples. The multi-class decision is made using the maximum score. In order to recreate the experiments in (Lucey *et al.*, 2010), the neutral examples are left out of the testing, resulting a forced-choice between the seven emotions. Again, the SVM's and logistic regressors are learned using leave-one-subject-out, that is, over a total of 123 trials.

2blue!15white

Table 5.4 Comparison of confusion matrices of emotion detection for the combined landmark (SPTS) and shape-normalised image (CAPP) features before and after identity normalisation. The average accuracy for all predicted emotions using the state of the art method in Lucey *et al.* (2010) is 83.27% (top table), using our method yields 95.21% (bottom table), a substantial improvement of 11.9%.

CK+ Emotion Recognition

Baseline SPTS+CAPP Lucey *et al.* (2010)

	An	Di	Fe	Ha	Sa	Su	Co
An	**75.0**	7.5	5.0	0.0	5.0	2.5	5.0
Di	5.3	**94.7**	0.0	0.0	0.0	0.0	0.0
Fe	4.4	0.0	**65.2**	8.7	0.0	13.0	8.7
Ha	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0
Sa	12.0	4.0	4.0	0.0	**68.0**	4.0	8.0
Su	0.0	0.0	0.0	0.0	4.0	**96.0**	0.0
Co	3.1	3.1	0.0	6.3	3.1	0.0	**84.4**

Identity Normalized SPTS+CAPP

	An	Di	Fe	Ha	Sa	Su	Co
An	**95.4**	0.0	0.0	0.0	0.0	0.0	4.7
Di	1.9	**94.2**	0.0	0.0	1.9	0.0	1.9
Fe	0.0	0.0	**84.2**	0.0	0.0	5.3	10.5
Ha	0.0	0.0	1.6	**96.9**	0.0	0.0	1.6
Sa	0.0	0.0	5.0	0.0	**90.0**	0.0	5.0
Su	0.0	0.0	0.0	0.0	0.0	**98.8**	1.2
Co	7.7	0.0	0.0	0.0	0.0	0.0	**92.3**

However, after combining scores from the SVM's using a logistic regression technique, the improvements are quite impressive. As in (Lucey *et al.*, 2010), results after the calibration of SVM scores indicate that the point data and image data capture different kinds of information. Combining these features yields impressive gains in performance. "Anger," especially seems to be improved using our method. In particular, only "Happiness" recognition appears to be reduced using our approach, but the reduction in hit rate is modest. Overall, this method gave a 95.21% average accuracy compared to the state of the art result of 83.27% reported in (Lucey *et al.*, 2010). Comparison of confusion matrices of emotion detection for the combined landmark (SPTS) and shape-normalized image (CAPP) features before and

after identity normalization is shown in Table 5.3.1.

5.3.2 Animation Control Experiments

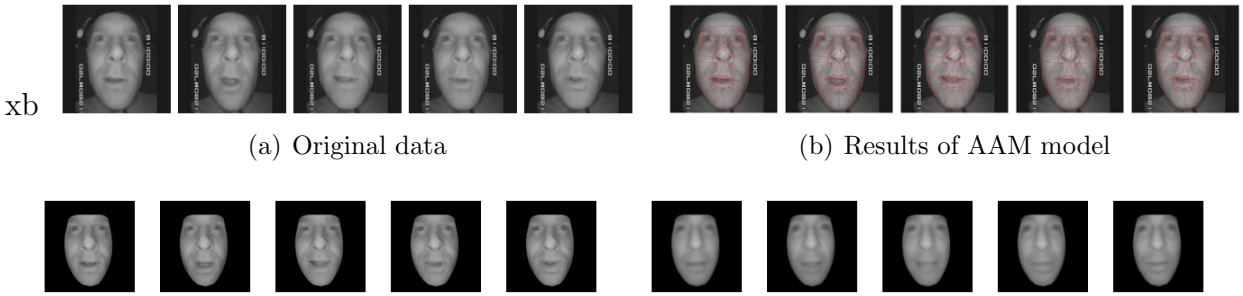
JAFFE

We again begin using the JAFFE dataset. Each image, after face detection and alignment, is labeled using a common 3D face mesh fitted with 27 bones created by an artist, as can be seen in Figure 5.2. Each bone is then positioned by an artist to a maximal and minimal position along a fixed path. This is proprietary software, but the method involves first associating each face mesh vertex with a non-renderable object called a "bone." Each vertex-bone dependency is weighted, so that bones have varying influence over face points – for example, a bone placed on the right upper eyelid might have 100% influence over the position of the eyelid vertices, but 0% influence on any other vertex. The bone position is then mapped to a particular function, such as translation, so that a x or y translation may cause a 50% influenced face vertice to move in the same direction, but only 50% of the distance. The result is a complex facial animation completely determined by movement of bones. To simplify animation, a bone or subset of bones are parameterized by a single scalar value, so that desired realistic non-rigid facial motions are re-created easily. Therefore, a blink movement can be pre-animated by bone position, where 0 represents a neutral state, 1, represents the eye fully open, and -1 represents the eye competely open. This animation is also a bone path. The position along the path is specified by a real value in $[-1, 1]$ as a fraction of the distance between the midpoint and the extreme values. Each JAFFE image is fully labeled using this software, yielding a labeled dataset of 27 scalar values between $[-1, 1]$.

We use MSE for evaluation purposes on the animation experiment, which corresponds to bone parameter recovery. In this case, we use linear regression as the predictive algorithm. The MSE reported is the average error in bone position for all test images. For each trial we leave one identity out from the trial, and compute both the MSE error and a prediction for the facial action. We compare our approach to PCA, using 30 and 100 dimensions. The results are summarized in Table 5.2. For the experiment labeled "none," the experiment denotes no unsupervised pre-processing step – the input is the raw image data. As it is difficult to gage the quality of the predictions from MSE alone, the predictions for a test subject using our method are shown in Figure 5.2, showing that the method does recover the facial actions produced by the subject quite well. The small standard error of the MSE indicates that, in general, the procedure is capable of predicting facial actions across all unseen subjects.

Marker-less Motion Capture Studio data

We now return to the challenging real-world problem of high quality facial animation control using video obtained from helmet cameras. This technique was used in the well known film Avatar and this data comes from Ubisoft, the company responsible for the brands Assassin's Creed and FarCry among others. FarCry 3 uses these technologies extensively. We obtained 16 videos of motion capture data without marker data. These videos are produced using infra-red cameras. Examples are shown in Figure 5.3.2. As is common in motion capture data, faces are captured using a helmet-mounted camera which reduces pose variation. The camera position is relatively fixed. However, this data presents new challenges due to variation stemming from the varied appearance of the actors. To compensate for appearance variation due to facial structure, an Active Appearance Model is applied to the data as shown in Figure 5.3.2, using 66 fiducial points. The resulting points derived from the model are used to warp each video frame to a common coordinate structure, removing pixels from outside the convex hull. These were used as inputs to our model, for a dataset totaling 3122 frames. Example outputs can also be seen in Figure 5.3(b).



(a) Original data (b) Results of AAM model

(c) Piecewise warped data (d) Identity normalization is applied

Figure 5.3 Motion capture training data using a helmet IR camera. Example pipeline for a test video sequence.

For our evaluations, we were given professional results of a bone-based model used as motion capture output. The bone positions are given as values in $[-1, 1]$ representing the relative distance from the neutral position for each bone along predefined space. Each experiment is run with one subject left out, and the results shown are the average MSE and standard error. Results of the experiments are shown in the right-most column of Table 5.2. We first show results using no preprocessing except for face detection using the Viola-Jones detector, cropped to a size 1.5 time the size of the detector to account for the distortion evident in the images. The results shown in Table 5.2 are per bone, and clearly some data processing must be used (None* refers to the images after shape normalization). Best results

were not obtained by applying PCA to the original data. The remaining experiments used shape-normalized data as inputs. As can be seen PCA improves the results, but the best results are achieved using our model. The classifier used was a simple linear regression, a more powerful discriminative classifier may be more effective.

5.4 Identity-Expression Active Appearance Model

In the preceding section, for many of the experiments, a key step is in the application of key-point detection algorithms, specifically the AAM. The AAM has been shown to work very well for subjects on which it has been trained while generalization to unseen identities has proven more difficult, due to the increased complexity of the shape model (Gross *et al.*, 2005). In the above experiments, the AAM is trained on subjects which appear in the test set. This is problematic because the features derived from the AAM seem to contribute the most to good results. If that is true, then an automatic expression recognition system must be allowed to train on the test subject as well, which raises the question of how good an automatic system can be, given that AAM's often perform poorly when not trained on the test subject.

Luckily the AAM contains a PCA model of the joint point and appearance data. This lends itself to a linear Gaussian interpretation which leads us to investigate whether key-point localization itself is improved using identity normalization. In this section we investigate identity normalization for key point detection.

5.4.1 Algorithm

The AAM fitting procedure we adapt is the inverse-compositional method as described by Matthews and Baker in (Matthews et Baker, 2004). We briefly review this method here. The modified objective in AAM fitting is to minimize the error between a template image, T and a set of warped images $I_j, j = 1, 2, ..., J$, which generally belong to the same identity. The modified objective is to minimize the squared error between the set of image over all image locations $\mathbf{x} = (x, y)$, by estimating parameters $\mathbf{p}_j$ of a warp function W.

$$\min_{\mathbf{p}_j} \sum_j \sum_{\mathbf{x}} [T - I_j(W(\mathbf{x}; \mathbf{p}_j))]^2 \tag{5.12}$$

We change notation here to make the following more clear. The warp function, W, can be interpreted as a map which determines a new location for any point $\mathbf{x}$, *i.e.*, $\mathbf{x}' = W(\mathbf{x}, \mathbf{p}_j)$. For AAM models, this is typically a piecewise affine transform based on the triangulation of a shape $\mathbf{s}$, which are represented by (x, y) point locations.

We shall quickly review how this optimization is performed with the classical AAM inverse compositional mapping approach, then replace the standard approach with our identity and expression factorization method and see the impact on the optimization procedure. In the standard AAM, $\mathbf{p}_j$ is the vector of weights for a set of eigen-shapes. The shape $\mathbf{s}$ is parameterized by a PCA model such that each $\mathbf{s}_j$ is explained by the mean shape $\mathbf{s}_0$ and a weighted combination of eigen-vectors $\mathbf{P_s}$ with weight parameters $\mathbf{p}_j$, such that $\mathbf{s}_j = \mathbf{s}_0 + \mathbf{P}_s\mathbf{p}_j$.

The inverse compositional approach is to optimize this objective by iteratively building up a series of warps to recover an optimal warp $W(\mathbf{x}, \mathbf{p}_j)$. Somewhat confusingly, both the template, T and the image I are warped. T is always warped from its initial shape $\mathbf{s}_0$. I, however, is warped to the current estimate $I(W(\mathbf{x}, \mathbf{p}_j))$. Once the optimal warp at the iteration is determined, $\Delta\mathbf{p}_j$, the current warp parameters $\mathbf{p}_j$ and $\Delta\mathbf{p}_j$ are "composed" in order to update the current warp parameters. When $\Delta\mathbf{p}_j$ is close to zeros, or if the warp is not changing much, the algorithm has converged. The current warp is formed by inverting the parameters of $\Delta\mathbf{p}_j$ (by negating) and applying the warp defined by these updated parameters to a warp composed of all previous warps. The modified objective at each iteration is then

$$\min_{\mathbf{p}_j} \sum_j \sum_{\mathbf{x}} [T(W(\mathbf{x}; \Delta\mathbf{p}_j)) - I_j(W(\mathbf{x}; \mathbf{p}_j))]^2 \tag{5.13}$$

Using a first order Taylor expansion, we have

$$\min_{\mathbf{p}_j} \sum_j \sum_x [T(W(\mathbf{x}; 0)) + \nabla T \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p}_j - I_j(W(\mathbf{x}; \mathbf{p}_j))]^2 \tag{5.14}$$

So that $\Delta\mathbf{p}_j$ can be given as

$$\Delta\mathbf{p}_j = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[\nabla T \frac{\partial W}{\partial \mathbf{p}}\right] [I_j(W(\mathbf{x}; \mathbf{p}_j)) - T(\mathbf{x})], \tag{5.15}$$

for an individual $\mathbf{p}_j$, where $\mathbf{H}$, the Hessian, can be written as

$$\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla T \frac{\partial W}{\partial \mathbf{p}}\right]^T \left[\nabla T \frac{\partial W}{\partial \mathbf{p}}\right] \tag{5.16}$$

The algorithm consists of pre-computing ∇T, $\frac{\partial W}{\partial \mathbf{v}_j}$, $\mathbf{H}_v, \mathbf{H}_w$ and inverses at the mean shape for efficiency. These are used along with Equation 5.15 to compute the change at $(\mathbf{p}_j = 0)$ required to minimize the error between the template and the current warped target. To adapt this algorithm for our purposes requires few changes.

In the AAM inverse compositional method, one must determine $W(\mathbf{x}, \mathbf{p}^k) = W(\mathbf{x}, \mathbf{p}^{k-1}) \circ$

$W(\mathbf{p};\Delta\mathbf{p})^{-1} \approxeq W(\mathbf{x},\mathbf{p}^{k-1}) \circ W(\mathbf{p};-\Delta\mathbf{p})$ at each iteration k. To compute the parameters of $W(\mathbf{x},\mathbf{p}^k)$ one then computes the corresponding changes to the current mesh vertex locations $\Delta\mathbf{s} = (\Delta x_1, \Delta y_1, \ldots, \Delta x_v, \Delta y_v)^T$, which are computed by composing the current warp.

The new parameters at time k are then given by $\mathbf{p}_j^k = \mathbf{P}_s^{-1}(\hat{\mathbf{s}}_j)$, where $\hat{\mathbf{s}}_j = \mathbf{s}_j^{k-1} + \Delta\mathbf{s}_j - \mathbf{s}_0)$. Now, in the case of our identity and expression decomposition model, we replace the PCA model with a factorized model with the form $\mathbf{s}_j = \mathbf{s}_0 + \mathbf{F}\mathbf{w} + \mathbf{G}\mathbf{v}_j$. The parameters are the vectors $\mathbf{w}$ and $\mathbf{v}_j$ belonging to the identity and the (expression of the person with that identity within) j^{th} image. In our identity and appearance decomposition approach, we now have an objective of the form,

$$\min_{\mathbf{w},\mathbf{v}_j} \sum_j \sum_{\mathbf{x}} [T(\mathbf{W}(\mathbf{x};\Delta\mathbf{w},\Delta\mathbf{v}_j)) - I_j(\mathbf{W}(\mathbf{x};\mathbf{w},\mathbf{v}_j))]^2 \tag{5.17}$$

Defining $\mathbf{z}_j = [\mathbf{w}^T, \mathbf{v}_j^T]^T$, Taylor series expansion yields

$$\min_{\mathbf{w},\mathbf{v}_j} \sum_j \sum_{\mathbf{x}} [T(\mathbf{W}(\mathbf{x};0,0)) + \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{z}_j}(\Delta\mathbf{z}_j) - I_j(\mathbf{W}(\mathbf{x};\mathbf{z}_j))]^2 \tag{5.18}$$

where $\Delta\mathbf{z}_j = [\Delta\mathbf{w}^T, \Delta\mathbf{v}_j^T]^T$. As identity and expression are assumed to be independent, this can be computed separately,

$$\Delta\mathbf{v}_j = \mathbf{H}_{\mathbf{v}}^{-1} \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{v}}\right] [I_j(\mathbf{W}(\mathbf{x};\mathbf{w},\mathbf{v}_j)) - T(\mathbf{x})]$$
$$\Delta\mathbf{w} = \mathbf{H}_{\mathbf{w}}^{-1} \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{w}}\right] [\frac{1}{J} \sum_j (I_j(\mathbf{W}(\mathbf{x};\mathbf{w},\mathbf{v}_j)) - T(\mathbf{x}))].$$

We then compute the warp composition as in (Matthews et Baker, 2004),

$$W(\mathbf{x},\mathbf{w}^k,\mathbf{v}_j^k) = W(\mathbf{x},\mathbf{w}^{k-1},\mathbf{v}_j^{k-1}) \circ W(\mathbf{x};-\Delta\mathbf{w},-\Delta\mathbf{v}_j).$$

However, computing the parameters $\mathbf{w}^k$ and $\mathbf{v}_j^k$ from $\Delta\mathbf{s}_j$ is no longer a simple matrix multiplication, as was the case when we updated $\mathbf{p}_j$ above, because $\mathbf{F}$ and $\mathbf{G}$ are not orthonormal and we use the sequence of images. Therefore, we use the expectation of Equation 5.11. That is, after computing Δs_j for each warp, let

$$\mathbf{d} \triangleq (\mathbf{w}^T, \mathbf{v}_1^T, \mathbf{v}_2^T, ..., \mathbf{v}_J^T)^T, \tag{5.19}$$

$$\hat{\mathbf{s}} \triangleq ((\mathbf{s}_1^{k-1} + \Delta\mathbf{s}_1 - \mathbf{s}_0)^T, ..., (\mathbf{s}_J^{k-1} + \Delta\mathbf{s}_J - \mathbf{s}_0)^T)^T \tag{5.20}$$

Then, with Ψ constructed as a diagonal matrix with Σ along the diagonal repeated J times,

$$\mathbb{E}[\mathbf{d}^k|\hat{\mathbf{s}}^k] = (\mathbf{A}^T\Psi\mathbf{A} + \mathbf{I}\sigma)^{-1}\mathbf{A}\Psi(\hat{\mathbf{s}}^k), \tag{5.21}$$

from which we can then compute the current warp $W(\mathbf{x}, \mathbf{w}^k, \mathbf{v}_j^k)$.

Texture variation is handled by alternating between training the model for texture variation and estimating the texture independently. Alternating between the two parameter optimization yielded the best results, although more efficient methods can be applied.

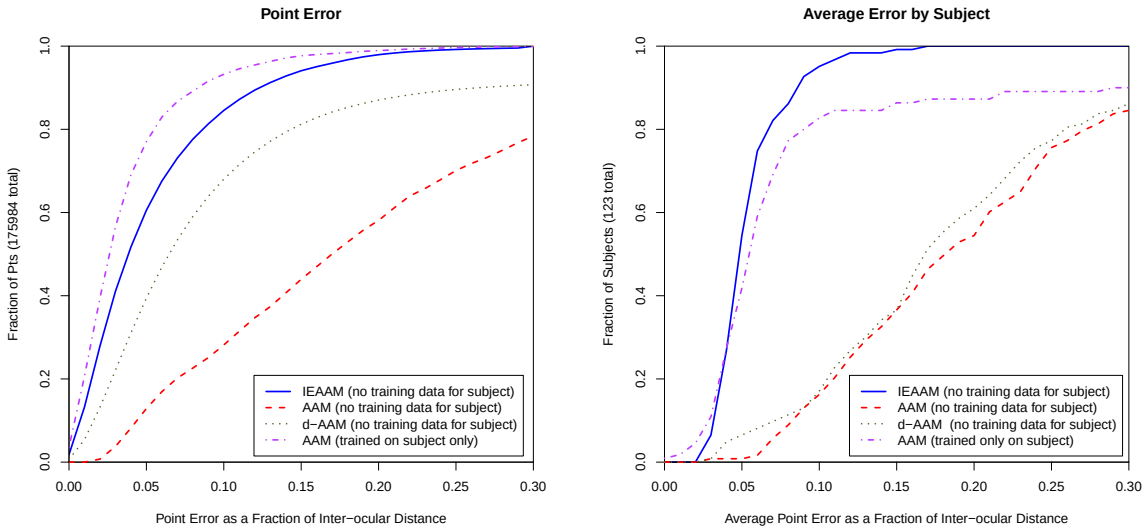
Evaluation

Since the CK+ dataset is also supplied with facial landmark annotations, we experimented using the Active Appearance Model extension described in previous sections. We used the same training set as described in the emotion recognition sections, training the factorized model on all but a single identity, and testing on those belonging to a single identity. For all our experiments, we use leave-one-subject-out cross-validation of a dataset of 2588 images and point locations. On average, this resulted in 2566 training images per experiment. Testing images were comprised of all remaining images from the CK+ dataset.

We evaluate our approach by adapting an existing software package, the ICAAM software package () implemented in Matlab, replacing the PCA point distribution model with our factorized model, and comparing the results against the original software. Although more complex AAM software is available (for example, the ICAAM package is not multi-resolution), our experiments are designed to illustrate both the effectiveness of the model and the simplicity of adapting existing approaches.

In order to achieve good results using this package, however, we used some pre-processing which improved ICAAM results : we first applied the OpenCV Viola-Jones frontal face-detector to remove large-scale pose variation and then aligned the images using a Procrustes analysis. The variation after Procrustes alignment was modeled using PCA with 3 components and added to the shape projection matrix in ICAAM in order to model pose. In our case, we added the pose parameters to the non-identity component matrix, $\mathbf{G}$ and then orthonormalization was applied. This follows Lucey *et al.*among others (Lucey *et al.*, 2010). ICAAM

was set to use the default 98% of shape variation, while our method used 200 components for identity and 50 for expression. Test images were initialized using the face-detector used for training.



(a) Cumulative error distribution.

(b) Subject average error, as fraction of inter-ocular distance, cumulative distribution.

Figure 5.4 Point-localization experiment evaluation, described in detail in text.

The results are summarized in Figure 5.4. In 5.4(a) we show the total cumulative distribution of error of all tested points for four methods. This analysis is widely used to evaluate key-point techniques (Matthews et Baker, 2004). Our method, the Identity-Expression AAM, (IEAAM), out-performs a discriminative AAM included in the Demolib distribution by Saragih *et al.*(Jason Saragih, 2009) (d-AAM) and the ICAAM AAM on which our INM is based. In both cases, the default settings are used, with PCA dimensionality chosen to account for 95% and 98% of variation. These three methods are not trained on the left-out subject, but on all remaining identities.

To provide another point of reference concerning what AAM performance is possible if training data is used for the subject of interest, we show the results of a subject-specific model using the AAM implementation provided in the DeMoLib distribution. This subject-specific AAM was trained on the sample of testing subject images described previously in Section 5.3.1, and tested on all remaining subject images from the CK+ dataset. This corresponds to using the first and last image, along with a sampling of intermediary images from each sequence as training images. In CK+, this corresponds to extreme expression poses. This configuration simulates the labeling of a range of motion image sequence, which can be

a good practical strategy for fitting an AAM with a small amount of labeling effort. The subject-specific AAM uses the project-out approach, (Matthews et Baker, 2004) and the inverse-compositional method. As expected, this method performs well, with less than 22.9% of predicted points being greater than 5% of the inter-ocular distance (calculated as the two farthest points on the eyes) from the ground truth.

We also compute the average inter-ocular distance error for each subject. In 5.4(b), we show the cumulative distribution of average subject error, which evaluates how well methods perform within each subject trial. The subject-trained AAM suffers from convergence errors, *i.e.*dramatic failures where all or many key-points completely falls off the face, which cause the average error for each subject to increase. In fact, the average error of all points for the subject-trained AAM is 23.3% of inter-ocular distance, while the average error of our method is 5.94%. Our method uses joint optimization on a sequence of frames adding stability to the point localization. Because of this, our method does not have these kinds of convergence errors, as shown in Figure 5.4(b),

Other methods, such as that of Van der Maaten and Hendriks report 4.69 pixel error using mixture modeling on the CK+ data (van der Maaten et Hendriks, 2010). They also report 0.23% convergence error rate at that level of accuracy. More crucially, however, test subjects in their testing regimen were not strictly prohibited from the training set. We are not aware of any AAM experiments using a large number of disparate subjects and testing on left-out subjects that report lower error. In any case, we are motivated more by the possibility of improving methods by using identity information, and believe that identity-normalization might easily be applied to any AAM-method. However, we do note that we see 0% convergence error.

These results strongly suggest that identity is an important source of information and there are measurable benefits to modeling such information explicitly. The average error for the CK+ set using the ICAAM code with face detection initialization is 24.36, with a standard deviation of 11.10, whereas our model yields an average pixel error of 7.15, with a standard deviation of 6.57. Both the bias and the variance is minimized simply by extricating the source of variation. The change to the algorithm, as discussed in previous sections, is relatively small. The result is significant improvement.

CHAPTER 6

CONCLUSION

This chapter concludes the thesis by summarizing both the contributions and limitations of the work, and finally presents some directions for future work.

As shown in the previous chapter, it is clear that identity labels provide an important source of information for expression recognition and other related tasks. By using identity labels in the method described above identity normalization can be used to improve the performance of supervised approaches to performance-driven animation and expression recognition tasks. Many models which are limited by the need for subject-specific training, which limits the quantity of labeled data which can be brought to bear, can be extended using the above approach. Clearly, approaches that work well for a single subject can be adapted, via unsupervised learning, to produce good quality results for unseen identities. In all cases, the results show that dealing with identity variation is an important aspect for expression-related tasks. In short, with regard to Hypothesis 3, identity information can and should be used to improve facial expression recognition.

Chapter 3, on the other hand, showed that using weak labels derived from video data combined with existing facial images can indeed aid learning tasks. Noisy labels are readily available in video for faces using this method. Adding a null-category with soft constraints produces better results than using the labeled data alone also improved performance. In all cases, the use of unlabeled data is able to improve the final classifier. The model is fully probabilistic and gives great flexibility. This included the use of sparsity-inducing priors and kernel methods. Going beyond experiments with a known noise level, using a realistic dataset and an estimate of the noise yielded improvement as well. On a realistic task of interest, improving a static face classifier using readily available video images, the method produced good results. Once again, this shows that the use of weakly labeled data can improve face recognition, as hypothesized.

Of course, the methods in Chapter 3 required much of the work in Chapter 2, which showed how combinations of hyper-priors could be used to obtain sparse kernel classifiers. Chapter 3 compared several sparse priors against a well-known data set, describing their properties and training algorithms. The Exponential hyper-prior was chosen because it had useful properties – sparsity and shrinkage. However, because it was a fully probabilistic method, this form of sparse kernel classification could easily be extended. One application of this, the Relevance Vector Random Field, is shown to improve performance on a medical segmentation problem.

This is just one example of possible extensions for the use of the Exponential prior.

In summary, the hope was to show that machine learning in vision should take into account useful information when possible. Labels, weak or not the target label, provide strong sources of information. For unlabeled data, the data gathering process can be structured in a way that provides weak labels, as in Chapter 3. On the other hand, labels are often provided, as in Chapter 4, which should be used as directly as possible. Although this information is not difficult to obtain, it is also not difficult to use, as shown in this thesis. However, the improvement in results indicates that the extra modeling effort is rewarding.

6.1 Limitations

In the case of identity-normalization, the main limitation is that identity and expression are assumed to be independent. This is, of course, generally false. Multi-linear analysis models this interaction as multiplicative and seems to be a better solution. As mentioned, the higher order SVD required for this analysis requires a full image tensor. There are methods which bypass this requirement, especially in the bilinear case, but it has not been applied to identity-normalization for performance-driven animation or key-point tracking. This would be the main direction for future research – adding a multiplicative interaction term for identity and expression to the linear model. However, it is also not clear that linear models are complex enough to capture realistic expression variation.

Of course, the work in Chapters 2 and 3 also has limitations. The extra modeling effort, and especially work with kernels, are computationally expensive, especially for very large training datasets. Unlike the SVM, our models scale in the training set size rather than the selection set of relevance vectors. However, in our case, training is easily parallelized, using iterative gradient descent techniques. The use of the GPGPU is able to achieve increases in performance which allow for comparable training times, when compared to SVM's. Sampling techniques can also be applied to the selection set, assuming that the number of relevance vectors necessary are fairly small. In the case of the K-CRF, greedy techniques can also be applied. However, it is generally the case that kernel methods will be computationally more expensive. A direction for further research are iterative step-wise techniques to reduce computational overhead, as in the fast RVM.

6.2 Future Work

The main direction for future work is toward better identity-normalization. This thesis has shown improved results on facial expression tasks using identity-dependent methods with identity-normalization as a pre-processing step. However, many models can integrate

this procedure within the learning method. For example, the convolutional neural net can be structured in a way to capture this information without pre-processing, as in Fasel (2002). A direction for future work is to integrate identity-normalization into more complex methods, such as the constrained local model, CLM Saragih *et al.* (2010), to combine a discriminative point-tracking system within an identity-independent representation. I plan to continue this work toward that goal in the future. Moreover, the linear models are extremely weak. It seems advantageous to increase model power by using a multi-resolution approach, as well as a hierarchical model *within* the expression-identity hierarchy. This would include using back-progation to model expression/identity weight vectors which are linear functions of other factors. This can be viewed as a deep-learning model. This is also a strong interest of mine for future work.

In regard to the weak label problem, I plan to continue to work toward more complex weak label data. For example, there is no reason why the weak label is required to be derived from the label proportion. At heart, the importance of the weak label is as a regularization term in the training of the discriminative classifier. That is, the classifier is regularized to respect the weak label distribution for unlabeled data. The label proportion is probably not the best estimate of the probability of the label being correct. The main direction is to investigate the role of better weak label distributions. In relation to this is extending the weak label framework to more complicated models, in particular, for structured prediction and other graphical models. There are many cases in which a very weak label is available which may help improve results, but which may also cause confusion when not handled properly. I plan to continue with this line of research as well.

REFERENCES

(2010). LFW : Results. http ://vis-www.cs.umass.edu/lfw/results.html.

(2010). VOC2010 preliminary results. http ://pascal-lin.ecs.soton.ac.uk/challenges/VOC/voc2010/results/index.html.

ABBOUD, B. et DAVOINE, F. (2004). Appearance factorization based facial expression recognition and synthesis. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* IEEE, vol. 4, 163–166 Vol.4.

AG, B. (2001). BioID face database. http ://www.bioid.com/support/downloads/software/bioid-face-database.html.

ATTIAS, H. (1999). Independent factor analysis. *Neural Computation*, 11, 803–851.

ATTIAS, H. (2000). A variational Bayesian framework for graphical models. *Advances in neural information processing systems*, 12, 209–215.

BACH, F. R. et JORDAN, M. I. (2005). A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley.

BARTLETT, M., LITTLEWORT, G., LAINSCSEK, C., FASEL, I. et MOVELLAN, J. (2004). Machine learning methods for fully automatic recognition of facial expressions and facial actions. *Systems, Man and Cybernetics, 2004 IEEE International Conference on.* vol. 1, 592–597 vol.1.

BARTLETT, M., MOVELLAN, J. et SEJNOWSKI, T. (2002). Face recognition by independent component analysis. *Neural Networks, IEEE Transactions on*, 13, 1450 – 1464.

BASU, S., BILENKO, M. et MOONEY, R. J. (2004). A probabilistic framework for semi-supervised clustering. *Semi-Supervised Learning*. 59–68.

BEAL, M. J. (2003). Variational algorithms for approximate Bayesian inference. Rapport technique.

BELHUMEUR, P., HESPANHA, J. et KRIEGMAN, D. (1996). Eigenfaces vs. fisherfaces : Recognition using class specific linear projection. B. Buxton et R. Cipolla, éditeurs, *Computer Vision . ECCV '96*, Springer Berlin / Heidelberg, vol. 1064 de *Lecture Notes in Computer Science*. 43–58. 10.1007/BFb0015522.

BENGIO, Y., DELALLAU, O. et ROUX, N. L. (2006). Label propagation and quadratic criterion. *Semi-Supervised Learning*, Olivier Chappelle, Bernhard Scholkopf, Alexander Zien.

BERG, T. L., BERG, A. C., EDWARDS, J., MAIRE, M., WHITE, R., TEH, Y., LEARNED-MILLER, E. et FORSYTH, D. A. (2004). Names and faces in the news. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*. IEEE Computer Society, Los Alamitos, CA, USA, vol. 2, 848–854.

BHOLE, C., PAL, C., RIM, D. et WISMÜLLER, A. (2013). 3d segmentation of abdominal ct imagery with graphical models, conditional random fields and learning. *Machine Vision and Applications*, 1–25.

BISHOP, C. et TIPPING, M. (2000). Variational relevance vector machines. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. 46–53.

BISHOP, C. *ET AL.* (2006). *Pattern recognition and machine learning*. Springer New York :.

BLACK, M. J. et YACOOB, Y. (1997). Recognizing facial expressions in image sequences using local parameterized models of image motion. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 25, 23—48.

BLANZ, V. et VETTER, T. (1999). A morphable model for the synthesis of 3d faces. ACM Press, 187–194.

BLUM, A. et MITCHELL, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the eleventh annual conference on Computational learning theory - COLT' 98*. Madison, Wisconsin, United States, 92–100.

BOYKOV, Y., VEKSLER, O. et ZABIH, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 2001.

BREIMAN, L. (2001). Random Forests. *Machine Learning*. 5–32.

BRESSON, X., VANDERGHEYNST, P. et THIRAN, J.-P. (2006). A Variational Model for Object Segmentation Using Boundary Information and Shape Prior Driven by the Mumford-Shah Functional. *International Journal of Computer Vision*, 68, 145–162.

BROWN, M., HUA, G. et WINDER, S. (2010). Discriminative Learning of Local Image Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

BRUNELLI, R. et POGGIO, T. (1993). Face recognition : features versus templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15, 1042 –1052.

BURGES, C. J. et PLATT, J. C. (2006). Semi-Supervised learning with conditional harmonic mixing. *Semi-Supervised Learning*, Olivier Chappelle, Bernhard Scholkopf, Alexander Zien.

CAMBRIDGE, A. L. (1994). ORL face database. Rapport technique, ATT Laboratories Cambridge. Published : http ://www.uk.research.att.com/facedatabase.html.

CAO, Z., YIN, Q., TANG, X. et SUN, J. (2010). Face recognition with learning-based descriptor. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. 2707–2714.

CARTEA, A. et HOWISON, S. (2003). *Distinguished limits of Levy-Stable processes, and applications to option pricing*. Mathematical Institute, University of Oxford.

CHANG, Y., VIEIRA, M., TURK, M. et VELHO, L. (2005). Automatic 3D Facial Expression Analysis in Videos. W. Zhao, S. Gong et X. Tang, éditeurs, *Analysis and Modelling of Faces and Gestures SE - 23*, Springer Berlin Heidelberg, vol. 3723 de *Lecture Notes in Computer Science*. 293–307.

CHENG, F., YU, J. et XIONG, H. (2010). Facial expression recognition in JAFFE dataset based on gaussian process classification. *Neural Networks, IEEE Transactions on*, 21, 1685–1690.

CHEON, Y. et KIM, D. (2009). Natural facial expression recognition using differential-AAM and manifold learning. *Pattern Recognition*, 42, 1340–1350.

CHHIKARA, R. et FOLKS, L. (1989a). *The inverse Gaussian distribution : theory, methodology, and applications*. CRC.

CHHIKARA, R. S. et FOLKS, L. (1989b). *The inverse Gaussian distribution : theory, methodology, and applications*. CRC.

COHEN, I., SEBE, N., GOZMAN, F., CIRELO, M. et HUANG, T. (2003). Learning Bayesian network classifiers for facial expression recognition both labeled and unlabeled data. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* IEEE Comput. Soc, vol. 1, I–595–I–601.

COOTES, T., EDWARDS, G. et TAYLOR, C. (2001). Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23, 681–685.

COOTES, T., TAYLOR, C., COOPER, D. et GRAHAM, J. (1995a). Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, 61, 38–59.

COOTES, T. F., TAYLOR, C. J., COOPER, D. H., GRAHAM, J. *ET AL.* (1995b). Active shape models-their training and application. *Computer vision and image understanding*, 61, 38–59.

CORDUNEANU, A. et JAAKKOLA, T. (2006). Data-Dependent regularization. *Semi-Supervised Learning*, Olivier Chappelle, Bernhard Scholkopf, Alexander Zien.

CORTES, C. et VAPNIK, V. (1995). Support vector network. *Machine Learning*, 20, 273–297.

COX, D. R. (1958). The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20, pp. 215–242.

CRAMER, J. S. (2003). *Logit Models from Economics and Other Fields*. Cambridge University Press, Cambridge.

DALAL, N. et TRIGGS, B. (2005). Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. vol. 1, 886–893 vol. 1.

DASARATHY, B. V. (1980). Nosing around the neighborhood : A new system structure and classification rule for recognition in partially exposed environments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 67–71.

DAVIS, J. V., KULIS, B., JAIN, P., SRA, S. et DHILLON, I. S. (2007). Information-theoretic metric learning. *Proceedings of the 24th international conference on Machine learning - ICML '07*. Corvalis, Oregon, 209–216.

DONATO, G., BARTLETT, M., HAGER, J., EKMAN, P. et SEJNOWSKI, T. (1999). Classifying facial actions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21, 974–989.

DRUCK, G., MANN, G. et MCCALLUM, A. (2008). Learning from labeled features using generalized expectation criteria. *SIGIR*. ACM, 595–602.

EDWARDS, G., LANITIS, A., TAYLOR, C. et COOTES, T. (1998). Statistical models of face images . improving specificity. *Image and Vision Computing*, 16, 203–211.

EKMAN, P. et ROSENBERG, E. L. (2005). *What the face reveals : basic and applied studies of spontaneous expression using the facial action coding system (FACS)*. Oxford University Press US.

ESSA, I., BASU, S., DARRELL, T. et PENTLAND, A. (1996). Modeling, tracking and interactive animation of faces and heads//using input from video. *Computer Animation '96. Proceedings*. 68–79.

ESSA, I. et PENTLAND, A. (1997). Coding, analysis, interpretation, and recognition of facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19, 757–763.

FASEL, B. (2002). Robust face analysis using convolutional neural networks. *Object recognition supported by user interaction for service robots*. IEEE Comput. Soc, vol. 2, 40–43.

FAUL, A. et TIPPING, M. (2002). Analysis of sparse bayesian learning.

FELLER, W. (1971). An introduction to probability theory and its applications. Vol. II. John Wiley & Sons. *Inc., New York-London-Sydney*.

FIGUEIREDO, M. *ET AL.* (2002). Adaptive Sparseness using Jeffreys' Prior. *Advances in Neural Information Processing Systems*, 1, 697–704.

FISHER, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 179–188.

FREY, B. J. et NEBOJSA, J. (1999). Estimating mixture models of images and inferring spatial transformations using the EM algorithm. Ft. Collins, CO, USA, 416–422.

GHAHRAMANI, Z. et BEAL, M. (2001). Graphical models and variational methods. M. Opper et D. Saad, éditeurs, *Advanced mean field methods : theory and practice*, MIT Press.

GOEL, N. et OF COMPUTER SCIENCE UNIVERSITY OF NEVADA, R. D. (2004). *Face Recognition : Experiments with Random Projection*. Citeseer.

GOUDEAUX, K., CHEN, T., WANG, S. et LIU, J. (2001). Principal component analysis for facial animation. *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*. vol. 3, 1501–1504.

GRANDVALET, Y. et BENGIO, Y. (2004). Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 529–536.

GROSS, R., MATTHEWS, I. et BAKER, S. (2005). Generic vs. person specific active appearance models. *Image and Vision Computing*, 23, 1080–1093.

GUILLAMET, D. et VITRIA, J. (2002). Non-negative matrix factorization for face recognition. M. Escrig, F. Toledo et E. Golobardes, éditeurs, *Topics in Artificial Intelligence*, Springer Berlin / Heidelberg, vol. 2504 de *Lecture Notes in Computer Science*. 336–344.

GUILLAUMIN, M., VERBEEK, J. et SCHMID, C. (2009). Is that you ? metric learning approaches for face identification. *Computer Vision, 2009 IEEE 12th International Conference on*. 498–505.

HANNES KRUPPA, M. C.-S. et SCHIELE, B. (2003). Fast and robust face finding via local context. *IEEE Workshop on Visual Surveillance and PETS*.

HASTIE, T. et TIBSHIRANI, R. (1995). Generalized Additive Models.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. et FRANKLIN, J. (2005). The elements of statistical learning : data mining, inference and prediction. *The Mathematical Intelligencer*, 27, 83–85.

HESTENES, M. R. et STIEFEL, E. (1952). Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49, 409–436.

HOCKING, R. (1976). The Analysis and Selection of Variables in Linear Regression. *Biometrics*, 32.

HOIEM, D., ROTHER, C. et WINN, J. (2007). 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation. *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

HONGCHENG WANG et AHUJA, N. (2003). Facial expression decomposition. *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 958–965 vol.2.

HOTELLING, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24, 417–441.

HUANG, G. B., JAIN, V. et LEARNED-MILLER, E. (2007a). Unsupervised joint alignment of complex images. *Proc. of ICCV*. Rio de Janeiro, Brazil, 153–160.

HUANG, G. B., JONES, M. J. et LEARNED-MILLER, E. (2008). Lfw results using a combined nowak plus merl recognizer. *Faces in Real-Life Images Workshop in ECCV*.

HUANG, G. B., RAMESH, M., BERG, T. et LEARNED-MILLER, E. (2007b). Labeled faces in the wild : A database for studying face recognition in unconstrained environments. Rapport technique 07-49, University of Massachusetts, Amherst.

JAAKKOLA, T. et HAUSSLER, D. (1999). Probabilistic kernel regression models.

JAAKKOLA, T. et JORDAN, M. (1997). A variational approach to Bayesian logistic regression models and their extensions. *Proceedings of the sixth international workshop on artificial intelligence and statistics*. Citeseer.

JASON SARAGIH, R. G. (2009). Learning AAM fitting through simulation. *Pattern Recognition*, 42, 2628–2636.

JAVIER, R. S., RODRIGO, V. et MAURICIO, C. (2009). Recognition of faces in unconstrained environments : a comparative study. *EURASIP Journal on Advances in Signal Processing*, 2009.

JOACHIMS, T. (2006). Transductive support vector machines. *Semi-Supervised Learning*, Olivier Chappelle, Bernhard Scholkopf, Alexander Zien.

JORDAN, M., GHAHRAMANI, Z., JAAKKOLA, T. et SAUL, L. (1999a). An introduction to variational methods for graphical models. *Machine learning*, 37, 183–233.

JORDAN, M. I. (2004). Learning in graphical models. 19, 140–155.

JORDAN, M. I., GHAHRAMANI, Z., JAAKKOLA, T. S. et SAUL, L. K. (1999b). An introduction to variational methods for graphical models. *Machine learning*, 37, 183–233.

KANADE, T., COHN, J. F. et TIAN, Y. (2000). Comprehensive database for facial expression analysis. *Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000. Proceedings*. IEEE, 46–53.

KINDERMANN, R., AUTHOR.) SNELL J. LAURIE (JAMES LAURIE), .-J. et SOCIETY, A. M. (1980). *Markov random fields and their applications / Ross Kindermann, J. Laurie Snell*. Providence, R.I. : American Mathematical Society.

KOLLER, D. et FRIEDMAN, N. (2009). *Probabilistic Graphical Models : Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

KOLMOGOROV, V. et ZABIH, R. (2004). What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 65–81.

KOTSIA, I. et PITAS, I. (2007). Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines. *IEEE Transactions on Image Processing*, 16, 172–187.

KRISHNAPURAM, B., CARIN, L., FIGUEIREDO, M. et HARTEMINK, A. (2005). Sparse multinomial logistic regression : fast algorithms and generalization bounds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27, 957–968.

KULIS, B., JAIN, P. et GRAUMAN, K. (2009). Fast similarity search for learned metrics. *IEEE PAMI*, 31, 2143–2157.

KUMAR, N., BERG, A., BELHUMEUR, P. et NAYAR, S. (2009). Attribute and simile classifiers for face verification. *Computer Vision, 2009 IEEE 12th International Conference on*. 365–372.

KUMAR, S. et HEBERT, M. (2006). Discriminative Random Fields. *International Journal of Computer Vision*, 68, 179–201.

LADES, M., VORBRUGGEN, J., BUHMANN, J., LANGE, J., VON DER MALSBURG, C., WURTZ, R. et KONEN, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *Computers, IEEE Transactions on*, 42, 300–311.

LAFFERTY, J., MCCALLUM, A. et PEREIRA, F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *In ICML*.

LAFFERTY, J., ZHU, X. et LIU, Y. (2004). Kernel conditional random fields. *Twenty-first international conference on Machine learning - ICML '04*. ACM Press, New York, New York, USA, 64.

LANITIS, A., TAYLOR, C. J. et COOTES, T. F. (1997). Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 743–756.

LAWRENCE, N. et JORDAN, M. (2005). Semi-supervised learning via Gaussian processes. *NIPS*, 17, 753–760.

LAWRENCE, N., PLATT, J. et JORDAN, M. (2005). Extensions of the Informative Vector Machine. J. Winkler, M. Niranjan et N. Lawrence, éditeurs, *Deterministic and Statistical Methods in Machine Learning*, Springer Berlin / Heidelberg, vol. 3635 de *Lecture Notes in Computer Science*. 56–87.

LEE, C.-S. et ELGAMMAL, A. (2005). Facial expression analysis using nonlinear decomposable generative models. *In IEEE International Workshop on Analysis and Modeling of Faces and Gestures*. Springer, 17–31.

LI, H., ROIVAINEN, P. et FORCHHEIMER, R. (1993). 3-D motion estimation in model-based facial image coding. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15, 545–555.

LIAO, P., SHEN, L., CHEN, Y. et LIU, S. (2004). Unified model in identity subspace for face recognition. *Journal of Computer Science and Technology*, 19, 684–690.

LIEN, J. J., KANADE, T., COHN, J. F. et LI, C. (1998). Subtly different facial expression recognition and expression intensity estimation. *1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998. Proceedings*. IEEE, 853–859.

LIU, W.-F., LU, J.-L., WANG, Z.-F. et SONG, H.-J. (2008). An Expression Space Model for Facial Expression Analysis. *2008 Congress on Image and Signal Processing*. IEEE, vol. 2, 680–684.

LOWE, D. G. (1999). Object recognition from local scale-invariant features. *Proc. of ICCV*. 1150–1157.

LUCEY, P., COHN, J. F., KANADE, T., SARAGIH, J., AMBADAR, Z. et MATTHEWS, I. (2010). The Extended Cohn-Kanade Dataset (CK+) : A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. IEEE, 94–101.

LYONS, M., AKAMATSU, S., KAMACHI, M. et GYOBA, J. (1998). Coding facial expressions with gabor wavelets. *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE, 200–205.

LYONS, M., BUDYNEK, J. et AKAMATSU, S. (1999). Automatic classification of single facial images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21, 1357–1362.

MANN, G. S. et MCCALLUM, A. (2007). Simple, robust, scalable semi-supervised learning via expectation regularization. *Proceedings of the 24th International Conference on Machine learning*. 600.

MATSUGU, M., MORI, K., MITARI, Y. et KANEDA, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16, 555–559.

MATTHEWS, I. et BAKER, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, 60, 135–164.

MCCALLUM, A., MANN, G. et DRUCK, G. (2007). Generalized expectation criteria. *Computer science technical note, University of Massachusetts, Amherst, MA*.

MCCALLUM, A., MANN, G. et DRUCK, G. (2007). Generalized expectation criteria. *UMass, Amherst, TR*.

MCFADDEN, D. L. (1984). Chapter 24 Econometric analysis of qualitative response models. Elsevier, vol. 2 de *Handbook of Econometrics*. 1395–1457.

METALLINOU, A., LEE, S. et NARAYANAN, S. (2008). Audio-Visual Emotion Recognition Using Gaussian Mixture Models for Face and Voice. *2008 Tenth IEEE International Symposium on Multimedia*. IEEE, 250–257.

MOGHADDAM, B., JEBARA, T. et PENTLAND, A. (2000). Bayesian face recognition. *Pattern Recognition*, 33, 1771–1782.

MOGHADDAM, B. et PENTLAND, A. (1997). Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19, 696 –710.

MOUSSOURIS, J. (1974). Gibbs and Markov random systems with constraints. *Journal of Statistical Physics*, 10, 11–33.

MPIPERIS, I., MALASSIOTIS, S. et STRINTZIS, M. (2008). Bilinear Models for 3-D Face and Facial Expression Recognition. *IEEE Transactions on Information Forensics and Security*, 3, 498–511.

MURPHY, K. P., WEISS, Y. et JORDAN, M. I. (1999). Loopy Belief Propagation for Approximate Inference : An Empirical Study. *In Proceedings of Uncertainty in AI*. 467–475.

NEAGOE, V.-E. et CIOTEC, A.-D. (2011). Subject-independent emotion recognition from facial expressions using a Gabor feature RBF neural classifier trained with virtual samples generated by concurrent self-organizing maps, 266–271.

NELDER, J. A. et WEDDERBURN, R. W. M. (1972). Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, 135, pp. 370–384.

NIGAM, K., MCCALLUM, A. et MITCHELL, T. (2006). Semi-supervised text classification using EM. *Semi-Supervised Learning*. 33–56.

NORDSTROM, M. M., LARSEN, M., SIERAKOWSKI, J. et STEGMANN, M. B. (2004). The IMM face database - an annotated dataset of 240 face images. Rapport technique, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby.

NOWAK, E. et JURIE, F. (2007). Learning visual similarity measures for comparing never seen objects. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*. 1–8.

OJALA, T., PIETIKÄINEN, M. et MÄENPÄÄ, T. (2002a). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE PAMI*, 971–987.

OJALA, T., PIETIKÄINEN, M. et MÄENPÄÄ, T. (2002b). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE PAMI*, 971–987.

OJALA, T., PIETIKÄINEN, M. et MÄENPÄÄ, T. (2002c). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE PAMI*, 971–987.

OTSUKA, T. et OHYA, J. (1997). Recognizing multiple persons' facial expressions using HMM based on automatic extraction of significant frames from image sequences. *Image Processing, International Conference on*. IEEE Computer Society, Los Alamitos, CA, USA, vol. 2, 546.

PANTIC, M. et ROTHKRANTZ, L. (2004). Facial action recognition for facial expression analysis from static face images. *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, 34, 1449–1461.

PANTIC, M., VALSTAR, M., RADEMAKER, R. et MAAT, L. (2005). Web-based database for facial expression analysis. *Proc. IEEE Int.l Conf. Multimedia and Expo*. 317–321.

PEARL, J. (1988). *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

PEARSON, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2, 559–572.

PENEV, P. S. et ATICK, J. J. (1996). Local feature analysis : A general statistical theory for object representation. *Network : Computation in Neural Systems*, 7, 477–500.

PENTLAND, A., MOGHADDAM, B. et STARNER, T. (1994a). View-based and modular eigenspaces for face recognition. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*. 84–91.

PENTLAND, A., MOGHADDAM, B. et STARNER, T. (1994b). View-based and modular eigenspaces for face recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*. IEEE Comput. Soc. Press, 84–91.

PHILLIPS, P. J., WECHSLER, H., HUANG, J. et RAUSS, P. J. (1998). The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16, 295–306.

PIGHIN, F., SZELISKI, R. et SALESIN, D. (1999). Resynthesizing facial animation through 3D model-based tracking. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. vol. 1, 143–150 vol.1.

PINTO, N., DICARLO, J. et COX, D. (2009). How far can you get with a modern face recognition test set using only simple features ? *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. 2591–2598.

PLATH, N., TOUSSAINT, M. et NAKAJIMA, S. (2009). Multi-class image segmentation using conditional random fields and global classification. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. ACM Press, New York, New York, USA, 1–8.

PRINCE, S., LI, P., FU, Y., MOHAMMED, U. et ELDER, J. (2011). Probabilistic models for inference about identity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP, 1.

QIAO, L., CHEN, S. et TAN, X. (2010). Sparsity preserving projections with applications to face recognition. *Pattern Recognition*, 43, 331–341.

REYNOLDS, J. et MURPHY, K. (2007). Figure-ground segmentation using a hierarchical conditional random field. *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)*. IEEE, 175–182.

RIM, D., HASAN, K. et PAL, C. (2011). Semi supervised learning for wild faces and video. *BMVC*.

SANDBACH, G., ZAFEIRIOU, S., PANTIC, M. et YIN, L. (2012). Static and dynamic 3D facial expression recognition : A comprehensive survey. *Image and Vision Computing*, 30, 683–697.

SANDERSON, C. et LOVELL, B. (2009). Multi-Region probabilistic histograms for robust and scalable identity inference. M. Tistarelli et M. Nixon, éditeurs, *Advances in Biometrics*, Springer Berlin / Heidelberg, vol. 5558 de *Lecture Notes in Computer Science*. 199–208. 10.1007/978-3-642-01793-3_21.

SARAGIH, J. M., LUCEY, S. et COHN, J. F. (2010). Deformable Model Fitting by Regularized Landmark Mean-Shift. *International Journal of Computer Vision*, 91, 200–215.

SEEGER, M. (2006). A taxonomy for Semi-Supervised learning methods. *Semi-Supervised Learning*, Olivier Chappelle, Bernhard Scholkopf, Alexander Zien.

SEUNG, H. S. (2000). COGNITION : The Manifold Ways of Perception. *Science*, 290, 2268–2269.

SHAN, C., GONG, S. et MCOWAN, P. W. (2009). Facial expression recognition based on local binary patterns : A comprehensive study. *Image and Vision Computing*, 27, 803–816.

SIBBING, D., HABBECKE, M. et KOBBELT, L. (2009). Markerless reconstruction of dynamic facial expressions. *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. 1778–1785.

SIBBING, D., HABBECKE, M. et KOBBELT, L. (2011). Markerless reconstruction and synthesis of dynamic facial expressions. *Computer Vision and Image Understanding*, 115, 668–680.

SIM, T., BAKER, S. et BSAT, M. (2002). The cmu pose, illumination, and expression (pie) database.

SINDHWANI, V., BELKIN, M. et NIYOGI, P. (2006). Geometric basis of Semi-Supervised learning. *Semi-Supervised Learning*, Olivier Chappelle, Bernhard Scholkopf, Alexander Zien.

SIROVICH, L. et KIRBY, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4, 519–524.

SPIEGELHALTER, D. J. et LAURITZEN, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20, 579–605.

SUTTON et MCCALLUM, A. (2010). An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning (FnT ML*.

TAIGMAN, Y., WOLF, L. et HASSNER, T. (2009). Multiple one-shots for utilizing class label information. *British Machine Vision Conference*. vol. 2.

TAN, H., CHEN, H. et ZHANG, J. (2009). <title>Estimating missing tensor data by face synthesis for expression recognition</title>. M. Rabbani et R. L. Stevenson, éditeurs, *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 72570D–72570D–6.

TASKAR, B., GUESTRIN, C. et KOLLER, D. (2003). Max-margin Markov networks. *In : Proc. Neural Information Processing Systems (NIPS)*.

TENENBAUM, J. B. et FREEMAN, W. T. (2000). Separating style and content with bilinear models. *NEURAL COMPUTATION*, 1247–1283.

TERZOPOULOS, D. et WATERS, K. (1993). Analysis and synthesis of facial image sequences using physical and anatomical models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15, 569–579.

THURSTONE, L. L. (1931). Multiple factor analysis. *Psychological Review*, 38, 406.

TIAN, Y., KANADE, T. et COHN, J. (2001). Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23, 97–115.

TIPPING, M. (2000a). The Relevance Vector Machine. *Advances in Neural Information Processing Systems*, 12.

TIPPING, M. (2000b). The Relevance Vector Machine. *Advances in Neural Information Processing Systems*, 12.

TIPPING, M. et BISHOP, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 61, 611–622.

TIPPING, M. E., TIPPING, M. E., FAUL, A. et AVENUE, J. J. T. (2003). Fast Marginal Likelihood Maximisation for Sparse Bayesian Models. *Proceedings Of The Ninth International Workshop On Artificial Intelligence And Statistics*, 3 – 6.

TISTARELLI, M. et NIXON, M. S., éditeurs (2009). *Advances in Biometrics*, vol. 5558 de *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

TONG, Y., LIAO, W. et JI, Q. (2007). Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE transactions on pattern analysis and machine intelligence*, 29, 1683–99.

TOUTENBURG, H. (1985). Everitt, B. S. : Introduction to Latent Variable Models. Chapman and Hall, London 1984. 107 pp., £ 9.50. *Biometrical Journal*, 27, 706.

TSALAKANIDOU, F. et MALASSIOTIS, S. (2010). Real-time 2D+3D facial action and expression recognition. *Pattern Recognition*, 43, 1763–1775.

TSOCHANTARIDIS, I., JOACHIMS, T., HOFMANN, T. et ALTUN, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal Of Machine Learning Research*, 6, 1453 – 1484.

TURK, M. et PENTLAND, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3, 71–86.

VALSTAR, M., JIANG, B., MEHU, M., PANTIC, M. et SCHERER, K. (2011). The first facial expression recognition and analysis challenge. *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. 921–926.

VALSTAR, M., PATRAS, I. et PANTIC, M. (2005). Facial action unit detection using probabilistic actively learned support vector machines on tracked facial point data. *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*. 76.

VAN DER MAATEN, L. et HENDRIKS, E. (2010). Capturing appearance variation in active appearance models. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. IEEE, 34–41.

VASILESCU, M. A. O. et TERZOPOULOS, D. (2002). Multilinear analysis of image ensembles : Tensorfaces. *In Proceedings Of The European Conference On Computer Vision*. 447–460.

VIOLA, P. et JONES, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proc. of CVPR*. 511–518.

WANG, X., LI, Z. et TAO, D. (2011). Subspaces indexing model on Grassmann manifold for image search. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 20, 2627–35.

WANG, Y., HUANG, X., LEE, C., ZHANG, S., LI, Z., SAMARAS, D., METAXAS, D., ELGAMMAL, A. et HUANG, P. (2004a). High resolution acquisition, learning and transfer of dynamic facial expressions. *Computer Graphics Forum*, 23, 677–686.

WANG, Y., HUANG, X., LEE, C.-S., ZHANG, S., LI, Z., SAMARAS, D., METAXAS, D., ELGAMMAL, A. et HUANG, P. (2004b). High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. *Computer Graphics Forum*, 23, 677–686.

WEIFENG LIU et YAN-JIANG WANG (2008). Expression feature extraction based on difference of Local Binary Pattern histogram sequences. *2008 9th International Conference on Signal Processing*. IEEE, 2082–2084.

WESTON, J. et WATKINS, C. (1999). Support Vector Machines for Multi-Class Pattern Recognition. *In : Proc. ESANN*.

WHITTAKER, J. (1990). *Graphical models in applied multivariate statistics*. Wiley series in probability and mathematical statistics : Probability and mathematical statistics. Wiley.

WHITTAKER, J. (2009). *Graphical Models in Applied Multivariate Statistics*. Wiley Publishing.

WINN, J., BISHOP, C. M. et JAAKKOLA, T. (2005). Variational message passing. *Journal of Machine Learning Research*, 6, 661–694.

WIPF, D. (2006). *Bayesian methods for finding sparse representations*. Thèse de doctorat, UNIVERSITY OF CALIFORNIA, SAN DIEGO.

WISKOTT, L., FELLOUS, J., KUIGER, N. et VON DER MALSBURG, C. (1997). Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19, 775–779.

WOLF, L., HASSNER, T. et TAIGMAN, Y. (2008a). Descriptor based methods in the wild. *Faces in Real-Life Images Workshop in ECCV*.

WOLF, L., HASSNER, T. et TAIGMAN, Y. (2008b). Descriptor based methods in the wild. *Faces in Real-Life Images Workshop in ECCV*. Citeseer.

WOLF, L., HASSNER, T. et TAIGMAN, Y. (2009a). The One-Shot similarity kernel. *Proc. of ICCV*.

WOLF, L., HASSNER, T. et TAIGMAN, Y. (2009b). Similarity scores based on background samples. *Proc. of ACCV*.

XU, Y., ZHONG, A., YANG, J. et ZHANG, D. (2010). LPP solution schemes for use with face recognition. *Pattern Recognition*, 43, 4165–4176.

YANG, G. et HUANG, T. S. (1994). Human face detection in a complex background. *Pattern Recognition*, 27, 53–63.

YEDIDIA, J. S., FREEMAN, W. T. et WEISS, Y. (2000). Generalized Belief Propagation. *IN NIPS 13*. MIT Press, 689–695.

YIN, L., CHEN, X., SUN, Y., WORM, T. et REALE, M. (2008). A high-resolution 3D dynamic facial expression database.

YUAN, M. et LIN, Y. (2007). On the nonnegative garrote estimator. *Journal of the Royal Statistical Society, Series B*, 69, 143–161.

YUILLE, A. L. (1991). Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3, 59–70.

ZENG, Z., PANTIC, M., ROISMAN, G. I. et HUANG, T. S. (2009). A survey of affect recognition methods : audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31, 39–58.

ZHANG, L., SNAVELY, N., CURLESS, B. et SEITZ, S. M. (2007). Spacetime faces : High-Resolution capture for˜Modeling and animation. Z. Deng et U. Neumann, éditeurs, *Data-Driven 3D Facial Animation*, Springer London, London. 248–276.

ZHANG, X. et GAO, Y. (2009). Face recognition across pose : A review. *Pattern Recognition*, 42, 2876–2896.

ZHANG, Y., MATUSZEWSKI, B. J., SHARK, L. et MOORE, C. J. (2008). Medical Image Segmentation Using New Hybrid Level-Set Method. *BioMedical Visualization, 2008. MEDIVIS '08. Fifth International Conference*. 71–76.

ZHANG, Z., LYONS, M., SCHUSTER, M. et AKAMATSU, S. (1998). Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multilayer perceptron. *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. 454–459.

ZHAO, W., CHELLAPPA, R., PHILLIPS, P. J. et ROSENFELD, A. (2003). Face recognition : A literature survey. *ACM Comput. Surv.*, 35, 399–458.

ZHOU, C. et LIN, X. (2005). Facial expressional image synthesis controlled by emotional parameters. *Pattern Recognition Letters*, 26, 2611–2627.

ZHOU, D. et SCHOLKOPF, B. (2006). Discrete regularization. *Semi-Supervised Learning*, Olivier Chappelle, Bernhard Scholkopf, Alexander Zien.

ZHU, J. et HASTIE, T. (2001). Kernel Logistic Regression and the Import Vector Machine. *Journal Of Computational And Graphical Statistics*, 14, 1081 – 1088.

ZHU, X. et GHAHRAMANI, Z. (2002). Learning from labeled and unlabeled data with label propagation.

ZHU, Y., DE LA TORRE, F., COHN, J. F. et ZHANG, Y. (2009). Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection.

ZHU, Z. et JI, Q. (2006). Robust Real-Time face pose and facial expression recovery. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*. IEEE Computer Society, Los Alamitos, CA, USA, vol. 1, 681–688.

APPENDIX A

Two sparse priors were discussed in Chapter 3, which represent novel algorithms. However, they were not relevant to the overall discussion. In this appendix, these algorithms are developed for the interested reader.

Mixtures of Gaussians Prior

The Mixture of Gaussian (MoG) prior can be seen as an extension of the Gaussian scale mixture to a discrete mixture. In the Gaussian scale mixture, the prior can be decomposed into

$$p(w_j) = \int N(w_j, \tau_j) p(\tau_j) d\tau_j \tag{A.1}$$

This can be restricted to a discrete model in which

$$p(w_j) = \sum_{k}^{K} \gamma_{jk} N(w_{jk}, \alpha_{jk}^{-1}), \sum_{k}^{K} \gamma_{jk} = 1 \tag{A.2}$$

Hence, the model is a two level hierarchical model where the visible output, $\mathbf{y}$ is determined by the logistic function of a linear combination of basis functions, and the weights used in the linear combination are distributed according to a mixture of Gaussians. This procedure is similar to that of independent factor analysis (IFA), a factorial mixture model first presented by Attias (Attias, 1999).

To generate $p(y)$, for N basis functions, choose a class or state variable s_j with probability γ_{jk}, then w_j from a Gaussian distribution given by α_{nk} and finally combine the basis functions $\mathbf{K}$, using the resulting $\mathbf{w}$, and finally pass this through the logistic function. Summing over all possible $s_j \in (1, 2, ..., K)$,

$$p(w_j) = \sum_{s_j} \gamma_{is_j} N(w_{is_j}, \alpha_{is_j}^{-1}) \tag{A.3}$$

The resulting multivariate prior $p(\mathbf{w})$ is also a mixture distribution.

$$p(\mathbf{w}) = (\sum_{s_1} N_{1s_1}\gamma_{1s_1})(\sum_{s_2} N_{2s_2}\gamma_{2s_2})...(\sum^{K} N_{ns_n}\gamma_{ns_n}) \tag{A.4}$$

$$= \sum_{s_1}\sum_{s_2}...\sum_{s_n} \gamma_{1s_1}\gamma_{2s_2}...\gamma_{ns_n}N_{1s_1}N_{2s_2}...N_{ns_n} \tag{A.5}$$

$$= \sum_{\mathbf{s}} \gamma_{\mathbf{s}} \prod_i N(w_j, \alpha_{js_j}^{-1}) \tag{A.6}$$

$$= \sum_{\mathbf{s}} \gamma_{\mathbf{s}} N(\mathbf{w}, \mathbf{A}_{\mathbf{s}}^{-1}). \tag{A.7}$$

The last is product of independent Gaussians, and therefore Gaussian and $\mathbf{A}_s = diag(\alpha_{js_j})$, and $\gamma_{\mathbf{s}} = \prod_i \gamma_{js_j}$. To summarize,

$$p(\mathbf{s}) = \gamma_{\mathbf{s}} \tag{A.8}$$

$$p(\mathbf{w}|\mathbf{s}) = N(\mathbf{w}, \mathbf{A}_{\mathbf{s}}^{-1}) \tag{A.9}$$

$$p(\mathbf{y}|\mathbf{w}) = \sigma(y_i\mathbf{K}_i\mathbf{w}) \tag{A.10}$$

$$p(\mathbf{s}, \mathbf{w}, \mathbf{y}) = \sigma(y_i\mathbf{K}_i\mathbf{w})N(\mathbf{w}, \mathbf{A}_{\mathbf{s}}^{-1})\gamma_{\mathbf{s}}. \tag{A.11}$$

The goal, then, is to maximize the posterior, $p(\mathbf{s}, \mathbf{w}|\mathbf{y})$, which is dependent on the parameters $\theta = (\alpha, \gamma)$. To do this, the Variational Bayesian approach discussed above can be adapted by introducing an approximating distribution q and minimizing the KL divergence between q and p. Again, this can be accomplished by minimizing $\mathcal{E} = -L(q)$.

The expectation of the complete log likelihood can be decomposed into three expectations corresponding to layers in the hierarchical model.

$$\mathcal{E} = -L = -\sum_{\mathbf{s}} \int_{\mathbf{w}} q(\mathbf{s}, \mathbf{w}|\mathbf{y}, \theta^{(t)}) \log \frac{p(\mathbf{s}, \mathbf{w}, \mathbf{y}|\theta)}{q(\mathbf{s}, \mathbf{w}|\mathbf{y}, \theta^{(t)})} d\mathbf{w} \tag{A.12}$$

$$= -\mathbb{E}[\log p(\mathbf{s})p(\mathbf{w}|\mathbf{s})p(\mathbf{y}|\mathbf{w})|\theta^{(t)}] + \mathbb{H}[q] \tag{A.13}$$

$$= -\mathbb{E}[\log p(\mathbf{s})|\gamma^{(t)}] - \mathbb{E}[\log p(\mathbf{w}|\mathbf{s})|\alpha^{(t)}] - \mathbb{E}[\log p(\mathbf{y}|\mathbf{w})] + \mathbb{H}[q] \tag{A.14}$$

Now we can treat each expectation separately. First

$$\mathbb{E}[\log p(\mathbf{s})|\gamma^{(t)}] = \sum_j \sum_{s_j} q(s_j|\mathbf{y}, \theta^{(t)}) \log \gamma_{js_j} \tag{A.15}$$

$$\mathbb{E}[\log p(\mathbf{w}|\mathbf{s})|\alpha_j] = \sum_j \sum_{s_j} q(s_j|\mathbf{y}, \theta^{(t)}) \int \log p(w_j|s_j) q(w_j|s_j, \mathbf{y}, \theta^{(t)}) \tag{A.16}$$

$$\mathbb{E}[\log p(\mathbf{y}|\mathbf{w})] = \int_{\mathbf{w}} \log p(\mathbf{y}, \mathbf{w}) q(\mathbf{w}|\mathbf{y}, \theta^{(t)}) \tag{A.17}$$

Finally, the entropy term

$$\mathbb{H}[q] = \sum_{\mathbf{s}} \int_{\mathbf{w}} q(\mathbf{s}, \mathbf{w}|\mathbf{y}, \theta^{(t)}) \log q(\mathbf{s}, \mathbf{w}|\mathbf{y}, \theta^{(t)}) d\mathbf{w} \tag{A.18}$$

In attempting to optimize these expectations, it becomes immediately clear that because of the form of the logistic, the resulting distributions, and their expectations are intractable. *E.g.*, in the expectation in (A.17), $\int \sigma(y\mathbf{K}_i\mathbf{w}) q(\mathbf{w}|\mathbf{y}, \theta) d\mathbf{w}$, where q represents a marginalization over the hidden states $\mathbf{s}$.

Thus again, the variational parameters ξ become useful,

$$\sigma(y_i\mathbf{K}_i\mathbf{w}) \geq g_i(y_i, \mathbf{K}_i, \mathbf{w}, \xi_i) = \sigma(\xi_i) \exp\big(\frac{y_i\mathbf{K}_i\mathbf{w} - \xi_i}{2} - \lambda(\xi_i)((y_i\mathbf{K}_i\mathbf{w})^2 - \xi_i^2)\big) \tag{A.19}$$

$$G(\mathbf{y}, \mathbf{K}, \mathbf{w}, \xi) = \prod^n g_i. \tag{A.20}$$

with

$$\mathcal{E}_G = -\sum_{\mathbf{s}} \int_{\mathbf{w}} q(\mathbf{s}, \mathbf{w}|\mathbf{y}, \theta^{(t)}) \log \frac{p(\mathbf{s}) p(\mathbf{w}|\mathbf{s}) G(\mathbf{y}, \mathbf{w}, \xi)}{q(\mathbf{s}, \mathbf{w}|\mathbf{y}, \theta^{(t)})} d\mathbf{w}. \tag{A.21}$$

Here, I have oomitted the dependence on θ to keep the notation uncluttered. Again q is an approximate distribution which factorizes instead of the true posterior. That is $q(\mathbf{s}, \mathbf{w}) = q_s(\mathbf{s}) q_w(\mathbf{w}|\mathbf{s})$. Now,

$$\mathcal{E}_G = -\sum_{\mathbf{s}} \int_{\mathbf{w}} q_s(\mathbf{s}) q_w(\mathbf{w}|\mathbf{s}) \log \frac{p(\mathbf{s}) p(\mathbf{w}|\mathbf{s}) G(\mathbf{y}, \mathbf{w}, \xi)}{q_s(\mathbf{s}) q_w(\mathbf{w}|\mathbf{s})} d\mathbf{w}. \tag{A.22}$$

To get $\log q_w^*$,

$$\mathcal{E}_G = -\int_{\mathbf{w}} q_w(\mathbf{w}) \sum_{\mathbf{s}} q_s(\mathbf{s}) \log \frac{p(\mathbf{s})p(\mathbf{w}|\mathbf{s})G(\mathbf{y},\mathbf{w},\xi)}{q_s(\mathbf{s})q_w(\mathbf{w}|\mathbf{s})} d\mathbf{w} \tag{A.23}$$

$$= -\int_{\mathbf{w}} q_w(\mathbf{w}|\mathbf{s})(\log p(\mathbf{w}|\mathbf{s}) + \log G(\mathbf{y},\mathbf{w},\xi)) + \text{const.}) + q_w(\mathbf{w}|\mathbf{s}) \log q_w(\mathbf{w}|\mathbf{s})] d\mathbf{w}, \tag{A.24}$$

and since the functional above is optimized when $\log q_w(\mathbf{w}|\mathbf{s}) = \log p(\mathbf{w}|\mathbf{s}) + \log G(\mathbf{y},\mathbf{w},\xi) +$ const.,

$$\log q_w^*(\mathbf{w}|\mathbf{s}) = -\frac{1}{2}(\mathbf{w}^T(\mathbf{A_s} + 2\sum(\lambda(\xi_i)\mathbf{K}_i^T\mathbf{K}_i))\mathbf{w} - \mathbf{w}^T\sum y_i\mathbf{K}_i^T) + \text{const.} \tag{A.25}$$

$$= \frac{1}{C}\exp(-\frac{1}{2}(\mathbf{w}^T(\mathbf{A_s} + 2\sum\lambda(\xi_i))\mathbf{K}_i^T\mathbf{K}_i))\mathbf{w} - \mathbf{w}^T(\sum y_i\mathbf{K}_i^T) \tag{A.26}$$

After exponentiating and normalizing,

$$q_w(\mathbf{w}|\mathbf{s}) = N(\mathbf{w} - \mathbf{m_s}, \mathbf{S_s}) \tag{A.27}$$

$$\mathbf{S_s} = (\mathbf{A_s} + 2\sum\lambda(\xi_i)\mathbf{K}_i^T\mathbf{K}_i)^{-1} \tag{A.28}$$

$$\mathbf{m_s} = \frac{1}{2}\mathbf{S_s}(\sum y_i\mathbf{K}_i^T) \tag{A.29}$$

Doing the same for $q(s)$, we have

$$\mathcal{E}_G = -\sum_s \int_{\mathbf{w}} q_w(\mathbf{w}|\mathbf{s})q_s(\mathbf{s}) \log \frac{p(\mathbf{s})p(\mathbf{w}|\mathbf{s})G(\mathbf{y},\mathbf{w},\xi)}{q_s(\mathbf{s})q_w(\mathbf{w}|\mathbf{s})} d\mathbf{w} \tag{A.30}$$

$$\log q_s^*(\mathbf{s}) = \log\gamma(s) - \frac{1}{2}\log|\mathbf{A_s}| - \frac{1}{2}(\langle\mathbf{w}|\mathbf{s}\rangle^T\mathbf{A_s}\langle\mathbf{w}|\mathbf{s}\rangle) \tag{A.31}$$

After exponentiating and normalizing,

$$q_s^*(\mathbf{s}) = \frac{\gamma(s)N(\langle\mathbf{w}|\mathbf{s}\rangle, \mathbf{A_s}^{-1})}{\sum_{\mathbf{s}'}\gamma(\mathbf{s}')N(\langle\mathbf{w}|\mathbf{s}'\rangle, \mathbf{A_{s'}}^{-1})}. \tag{A.32}$$

This results in a mixture distribution, where $\langle\mathbf{w}|\mathbf{s}\rangle = \int_{\mathbf{w}} \mathbf{w}q_w(\mathbf{w}|\mathbf{s})d\mathbf{w} = \mathbf{m_s}$. Setting derivatives to 0 to find updates for each of the parameters, γ, α, ξ, maximization of $\langle\log p(\mathbf{s})\rangle$ is subject to the constraint that γ sum to unity for $\mathbf{s}$.

$$\max \sum_s q_s(\mathbf{s}) \log \gamma(\mathbf{s}) - \lambda\Big(\sum_s \gamma(\mathbf{s}) - 1 \Big) = \max \sum_j \sum_k \gamma(s_{jk}) q_s(s_{jk}) \log \gamma(s_{jk}) - \lambda_j(\sum_k \gamma(s_{jk}) - 1), \tag{A.33}$$

and

$$\gamma(s_{jk}) = q_s(s_{jk}). \tag{A.34}$$

For α, taking derivatives, and setting to 0, yields the update

$$\alpha_{jk} = \frac{q_s(s_{jk})}{q_s(s_{jk})(\langle w_j^2 | \mathbf{s}_{jk} \rangle)}. \tag{A.35}$$

To maximize ξ, we have

$$\max_{\xi_i} \log \sigma(\xi_i) + \frac{1}{2}(y_i \mathbf{K}_i \langle \mathbf{w} \rangle - \xi_i) - \lambda(\xi_i)(\mathbf{K}_i^T \langle \mathbf{w}\mathbf{w}^T \rangle \mathbf{K}_i - \xi_i^2) \tag{A.36}$$

$$\frac{\partial}{\partial \xi_i} = \frac{1}{2} - \sigma(\xi_i) - \lambda^{'}(\xi_i)(\mathbf{K}_i^T \langle \mathbf{w}\mathbf{w}^T \rangle \mathbf{K}_i - \xi_i^2) - 2\xi_i \lambda(\xi) \tag{A.37}$$

Noting that $\lambda(\xi_i)$ can be rewritten as

$$\lambda(\xi_i) = \frac{1}{2\xi_i}\big(\sigma(\xi_i) - \frac{1}{2}\big), \tag{A.38}$$

and that ξ_i is maximized when

$$\lambda^{'}(\xi_i)(\mathbf{K}_i^T \langle \mathbf{w}\mathbf{w}^T \rangle \mathbf{K}_i - \xi_i^2) = 0, \tag{A.39}$$

since $\lambda^{'}$ is a strictly positive function, the M-step update for ξ_i is

$$\xi_i^2 = \mathbf{K}_i^T \langle \mathbf{w}\mathbf{w}^T \rangle \mathbf{K}_i. \tag{A.40}$$

In this case, $\langle \mathbf{w}\mathbf{w}^T \rangle = \sum_{\mathbf{s}} q_s(\mathbf{s}) \mathbf{S}_{\mathbf{s}} + \sum_s q_s(\mathbf{s}) \mathbf{m}_{\mathbf{s}} \mathbf{m}_{\mathbf{s}}^T$. Individual weight averages can be

found by summing out over joint distributions as follows :

$$p(s_{jk}) = \sum_{\{\mathbf{s}|s_{jk}\notin\mathbf{s}\}} p(\mathbf{s}) \tag{A.41}$$

$$p(s_{jk})\langle w_j|s_{jk}\rangle = \sum_{\{\mathbf{s}|s_{jk}\notin\mathbf{s}\}} p(\mathbf{s})\langle w_j|\mathbf{s}\rangle \tag{A.42}$$

$$p(s_{jk})\langle w_j^2|s_{jk}\rangle = \sum_{\{\mathbf{s}|s_{jk}\notin\mathbf{s}\}} p(\mathbf{s})\langle w_j^2|\mathbf{s}\rangle \tag{A.43}$$

The $\mathcal{E}_G$ is computed by summing the following :

$$-\langle \log G(\mathbf{y},\mathbf{w},\xi)\rangle = -\sum_i \Big\{ \log\sigma(\xi_i) - \frac{1}{2}(y_i\mathbf{K}_i\langle\mathbf{w}\rangle - \xi_i) + \lambda(\xi_i)(\mathbf{K}_i^T\langle\mathbf{w}\mathbf{w}^T\rangle\mathbf{K}_i - \xi_i^2)\Big\} \tag{A.44}$$

$$-\langle \log p(\mathbf{w}|\mathbf{s})\rangle = -\frac{1}{2}\sum_s q(\mathbf{s})\log|\mathbf{A_s}| + \frac{1}{2}\sum_s q(\mathbf{s})\langle\mathbf{w}|\mathbf{s}\rangle^T\mathbf{A_s}\langle\mathbf{w}|\mathbf{s}\rangle \tag{A.45}$$

$$-\langle \log p(\mathbf{s})\rangle = -\sum_{\mathbf{s}} q(\mathbf{s})\log\gamma(\mathbf{s}) \tag{A.46}$$

$$\langle \log q(\mathbf{w}|\mathbf{s})\rangle = \frac{1}{2}\sum_s q(\mathbf{s})\log|\mathbf{S_s}| - \frac{1}{2}\sum_s q(\mathbf{s})\mathbf{m}_\mathbf{s}^T S_\mathbf{s}\mathbf{m}_\mathbf{s} \tag{A.47}$$

$$\langle \log q(\mathbf{s})\rangle = \sum_i\sum_k \gamma(s_{jk})\log\gamma(s_{jk}) \tag{A.48}$$

The predictive distribution is approximated, again, by using the expected $\langle\mathbf{w}\rangle$, in order to reap the benefit of sparsity.

The problem is that for even small datasets with a small number of classes, there is an exponential explosion in the size of the states. For instance with 200 datapoints and 2 states, 2^{200} sums to compute. However, under $k = 1$, we have a non-informative uniform hyperprior, which approximates the Jeffrey's prior. Figure (A.1) are results for a very small dataset of 20 points and 2 classes. Here we see that it has regularized itself well, even in the case of a small dataset. Figure (A.2) shows the classifier overlaid on the training set. Predictably, the results are not very good considering the small training set, which took roughly 40 minutes to produce. A direction for further research is the use of GPU for computing the enormous number of sums.

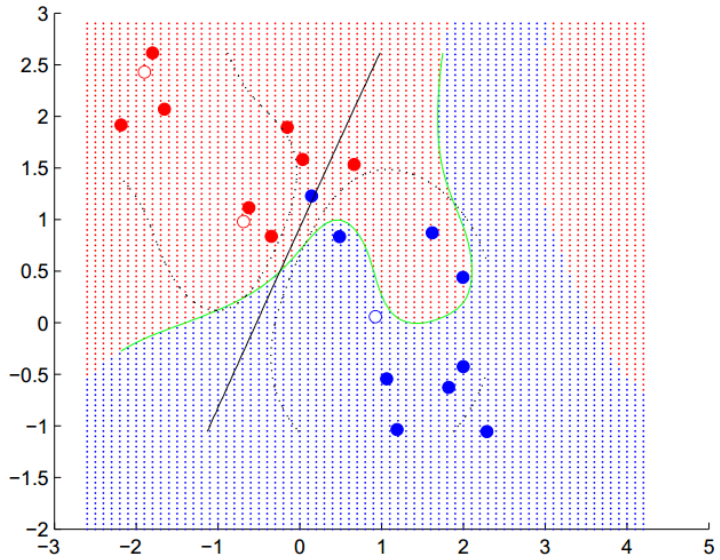


Figure A.1 MoG prior plot, $K = 2, n = 20$

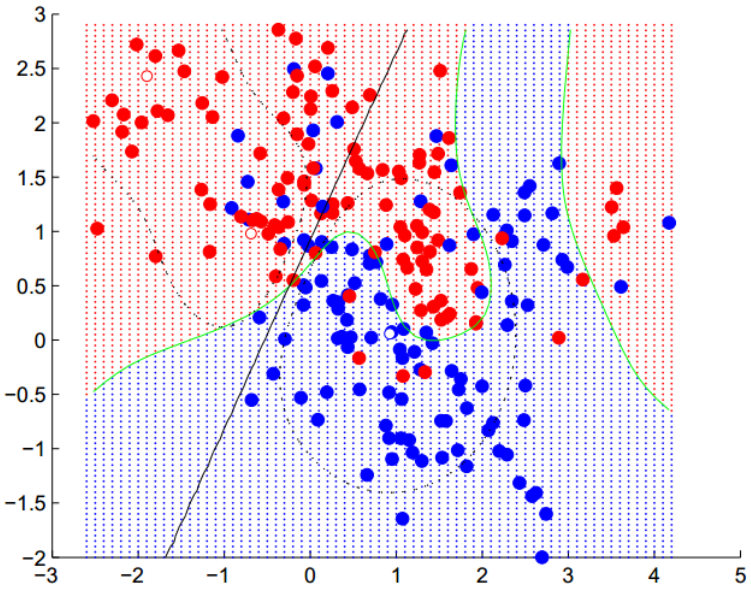


Figure A.2 MoG prior plot, $K = 2, n = 20$ on full training set

Figure (A.1) shows the results of a uniform prior, where each class basically collapses. These give very good results, although some indications of over fitting, where the flexibility

of the true MoG prior may be relevant.

Table A.1 Best Results for MoG

	training error	testing error	Bayes error	NSV	γ
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
MoG	.145	.205	.19	96	-

Non-negative Garrote

The non-negative garrote is a procedure introduced by Breiman in 1995 (Breiman, 2001). This procedure is motivated by classical statistics, and is quite similar to Least Angle Regression (LARS) and older methods such as stage-wise regression. In this case, the regression proceeds along a solution path, ending when the projected decrease in model error falls below a threshold. In essence, the non-negative garrote (NNG) procedure falls somewhere between ridge regression and the LASSO, by penalizing weights as a function of their magnitude, as in ridge regression but not in a quadratic smooth sense which does not lead to sparsity. The sparsity is induced by a penalty term on the absolute size of the weight, but as a function of its weight, rather than constant as in the l_1 norm. Although the non-negative garrote is not a Bayesian procedure, the regularization penalty can be viewed as a prior of the form in Table (3.1) by exponentiation and normalization.

In this section, I introduce the algorithm as originally presented by Breiman in the context of linear regression (Breiman, 2001) before extending this procedure to KLR.

Let y_n be distributed $N(\mathbf{x}_n^T\mathbf{w}, \tau\mathbf{I})$, where τ is the variance of the residual $\mathbf{y} - \mathbf{X}\mathbf{w}$, where $\mathbf{X}$ is the design matrix $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)^T$. Then the ML estimate of $\mathbf{w}$ is the OLS estimate

$$\hat{\mathbf{w}} = \text{argmin}_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{n} (\mathbf{y}_i - \sum_{j=1}^{d} w_j \mathbf{X}_{ij})^2 \tag{A.49}$$

The key to the non-negative garrote is a cost term for each weight component – $c_j \hat{w}_j$, where $\hat{w}_j$ is the OLS estimate.

$$\begin{aligned} \min_{\mathbf{c}} &= \frac{1}{2} \sum_i (y_i - \sum_j c_j \hat{w}_j \mathbf{X}_{ij})^2 \\ &\text{subject to } \sum_j c_j \leq s \\ &\forall j,\ c_j \geq 0. \end{aligned} \tag{A.50}$$

The s term is called the garrote, as it "caps" the weights to remain under a certain amount, similar to the slack penalty in the SVM. Using Lagrange multipliers, this results in

$$\min_{\mathbf{c}} \max_{\lambda,\tau} \frac{1}{2} \sum_i (y_i - \sum_j c_j \hat{w}_j \mathbf{X}_{ij})^2 + \lambda(s - \sum_j c_j), \text{ subject to } \forall j,\ c_j \geq 0. \tag{A.51}$$

The MAP estimate is then given by $\mathbf{w}^{\text{nng}} = (c_j \hat{w}_j)$. Thus λ represents a penalty on the negative log likelihood where c_j is a function of both the size of $\hat{w}_j$ as well as error $\mathbf{y} - \mathbf{X}\mathbf{w}^{\text{nng}}$. This leads to a sparse solution which is closer to the l_0 norm.

Although it is possible to select a given size for λ, and compute an estimate under this model, in practice, the general idea is to select a path for $\lambda_i \in (\lambda_1, \lambda_2, ...)$, and solve the quadratic problem for each, until the residual error falls below a threshold, as in (Breiman, 2001).

However, it has been noted by Yuan and Lin (Yuan et Lin, 2007), that the solution path is piecewise linear, as is clear by inspection. This being the case it should be possible to start with an empty model, and gradually add components with step sizes determined by the lambda necessary for the next candidate weight to be added to the model. When either the model residual is low or the necessary step size is too high, the procedure terminates. Their algorithm, very similar to LARS, is presented below.

First to simplify notation, $\sum_i (y_i - \sum_j c_j \hat{w}_j \mathbf{X}_{ij})$ can be rewritten as $\mathbf{y} - \mathbf{Z}\mathbf{c}$, where $\mathbf{Z} = (\mathbf{X}_1 w_1, \mathbf{X}_2 w_2, ..., \mathbf{X}_d w_d)^T$, where $\mathbf{X}_j$ is the j^{th} column of $\mathbf{X}$. We will denote $(\mathbf{X}_j w_j) = \mathbf{Z}_j$.

1. Initialize $\mathbf{c} = \mathbf{0}, \mathbf{r} = \mathbf{y}$, and let $\mathbf{m} = \text{argmax}_{\ k} \mathbf{Z}_k^T \mathbf{y}$, be the initial model (an index set).
2. Now we need to determine the direction in which the residual error lies in the subspace of the current model. Let $\gamma = \mathbf{0}$.

$$\gamma_{\mathbf{m}} = (\mathbf{Z}_{\mathbf{m}}^T \mathbf{Z}_{\mathbf{m}})^{-1} \mathbf{Z}_{\mathbf{m}} \mathbf{r} \tag{A.52}$$

   The direction is necessary in order to determine how far the current model can progress along γ. We note that γ is an n dimensional vector whose values are 0, unless the i^{th} component is contained in the model.

3. Given the direction, is there a step size α_j such that the residual error after progressing in $\alpha_j \gamma$ will lead to $\mathbf{Z}_j$ being added to the set ? The solution to this question is given by solving for α_j for each $j \notin \mathbf{m}$,

$$\mathbf{Z}_j^T (r - \alpha_j \mathbf{Z}\gamma) = \mathbf{Z}_k^T (r - \alpha_k \mathbf{Z}\gamma) \tag{A.53}$$

Where k is any element of $\mathbf{m}$.

4. Given the direction, is there a step size α_j such that after progressing in $\alpha_j\gamma$, $\mathbf{Z}_j$ will be dropped from the set ? Compute

$$\alpha_j = \min(\beta_j, 1) \tag{A.54}$$

$$\beta_j = -c_j/\gamma_j \tag{A.55}$$

for all $j \in \mathbf{m}$. If α_j is positive, this measures how much $s = \sum c_j$ must be decreased to drop j from the active model.

5. If all $\alpha_j \leq 0$, then we can not progress any further, so report convergence.
6. If $\min_{j:\alpha_j>0} \alpha_j > 1$, then we can only progress by decreasing $\mathbf{s}$ to drop all j from the current set, so report convergence.
7. Otherwise, let $\alpha_k = \min_{j:\alpha_j>0} \alpha_j$. Set $\mathbf{c} = \mathbf{c} + \alpha_k\gamma$. If $k \notin \mathbf{m}$, add k to $\mathbf{m}$. Otherwise, remove k from $\mathbf{m}$. Iterate, until convergence.

The non-negative garrote has path consistency results based on the assumption that the initial estimate $\hat{w}$ is consistent with a slower rate of convergence to consistent estimates (which it is if the initial estimate is the OLS) (Yuan et Lin, 2007). This is in contrast to the LASSO for which path consistency is assumed under more general assumptions concerning $\mathbf{w}$, which cannot be tested since nothing about $\mathbf{w}$ can be known a priori.

Of course, under kernel logistic regression, the same idea of residual error does not exist, although the above procedure can be extended by viewing the logistic regression as linear regression under the logit link function, $\sigma(a) = (1 + \exp(-a))^{-1}$. That is, the algorithm first estimates $\hat{w}$, and performs the same steps against the targets $z_i = \mathbf{K}_i\mathbf{w}$. The algorithm must then be modified slightly to take into account the heteroskedacity of the target residuals, which are given by $\sigma(\mathbf{K}_i\mathbf{w})(1 - \sigma(\mathbf{K}_i\mathbf{w}))$, and so we use the linear approximation given by

$$\hat{y} = \mathbf{K}_i\hat{\mathbf{w}} + \frac{y_i(1 - \sigma(\mathbf{K}_i\mathbf{w}))}{(\sigma(\mathbf{K}_i\mathbf{w}))(1 - \sigma(\mathbf{K}_i\mathbf{w}))} \tag{A.56}$$

$$\hat{\mathbf{w}} = \text{argmin} - \sum \log \sigma(-y_i\mathbf{K}_i\mathbf{w}) + \frac{1}{2}\lambda\mathbf{w}^T\mathbf{K}\mathbf{w}. \tag{A.57}$$

λ is required here to keep $\mathbf{w}$ small. The results for the non-negative garrote are shown in figure (A.3) and figure (A.2).

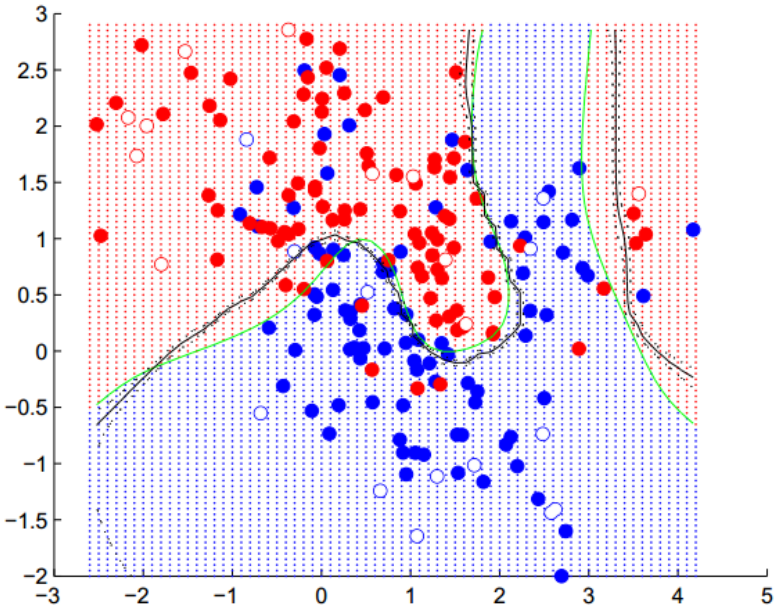


Figure A.3 Non-negative Garrote classifier

Table A.2 Results for Non-Negative Garrote

	training error	testing error	Bayes error	NSV	γ
SVM	.150	.220	.19	130	-
RVM	.160	.215	.19	12	-
NNG	.165	.220	.19	24	-

It is well known that without regularization logistic regression in a linearly separable space will lead to infinite $\mathbf{w}$. As such there is nothing to stop the c_j from also approaching infinity, as made clear in the figures. The probability becomes a step function as the garrote increases, with no uncertainty. Adding penalty terms to the procedure is a direction for future research.